MindSpore 大模型报错解决地图

目 录

1 MindSpore 常见大模型问题和解决方案

MindSpore Transformers 套件目标是构建一个大模型训练、微调、评估、推理、部署的全流程开发套件，是基于 MindSpore 内置的并行技术和组件化设计。本帖根据使用 MindSpore 大模型过程中积累的经验，总结了包含报错信息和解决方案的常见案例（**当前仅涵盖迁移过程中的问题，持续更新中**），可根据报错信息找到对应的解决方案。

2 环境问题案例

2.1 MindSpore 和 Flash attention 特性的适配问题

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.1

执行模式： 不限

2. 报错信息

2.1 问题描述

使用低于 2.2.10 版本的 MindSpore 会出现如下报错：

```
File "/opt/mindformers/mindformers/models/gpt bigcode/gpt_ bigcode_modules .py",
line 443, in __init__parallel config=attention parallel config)

File "/opt/mindformers/mindformers/models/gpt bigcode/gpt bigcode modules.py", line
126, in init  self.flash attention= nn.FlashAttention (self.size per head,
attention_dropout_rate, prev_block_num=65536,

AttributeError: module 'mindspore.nn'has no attribute  'FlashAttention'
```

3. 根因分析

低版本 MindSpore 不支持 flash attention（如 MindSpore2.0beta、MindSpore2.1），需要升级高版本才能支持这个特性。

4. 解决方案

1. 当在使用低版本 MindSpore 出现 flash attention 报错时（如 MindSpore2.0beta、MindSpore2.1），可以先改用常规 self-attention 替换 flash attention 算法，从而规避报错。

2. 使用 MindSpore2.2.10 版本。

2.2 MindSpore 保存模型出现维度不对应问题

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.1

执行模式： 不限

2. 报错信息

2.1 问题描述

使用 Ascend+MindSpore2.1 环境，将并行设置为 dp:mp:pp=2:2:2 时，每次保存模型会报如下信息（有时候是报 WARNING）：

```
[WARNING] MD(1764057, ffff9bd28b20, python):2023-09-08-20: 30: 46.738.589
[mindspore/ccs rc/minddata/dataset/engine/da tasetops/data_queue_op.cc:115]
~DataQueueop]  preprocess_batch: 600; batch_queue: 8, 8, 8, 8, 8, 8, 8, 8, 8, 8;

push_start_time -> push_end_time 2023-09-08-20:29:44.203.784 -08-09-2023 ı
20:29:44.204.298   2023-09-08-20:29:44.204.457 -> 2023-09-08-20:29:44.204.748

2023-09-08-20: 29:53.230.823 -> 2023-09-08-20:29:53.231.299

2023-09-08-20:29:53. 231.463 -> 2023-09-08-20:29:53.231.726

2023-09-08-20:30:02.229.866 -> 2023-09-08-20:30:02.230.527  2023-09-08-
20:30:02.230.696 ->  2023-09-08-20:30:02.231.047
```

```
2023-09-08-20:30:11.231.599 -> 2023-09-08-20:30:11.232.006

2023-09-08-20:30: 11.232.116 -> 2023-09-08- 20:30:11.232.359

2023-09-08-20:30: 20.206.672 -> 2023-09-08-20: 30:20.207.262

2023-09-08-20:30:20.207.400 - 2023-09-08-20:30:20.207.788    For more details,
please refer to the FAQ at https://www.mindspore.cn/docs/en/master/fag/data
processing.html.

Traceback (most recent call last):  File "wizardcoder/run_ wizardcoder.py", line
149, in <module>    device_id=args.device_ id)  File
"wizardcoder/run_wizardcoder.py"  line 81, in main
task.train(train_checkpoint=ckpt.    resume=resume)    File
"/home/wizardcoder/1_wizardcoder-m indfomers/mindfomers/trainer/t rainer.py", line
424, in train    is full config=True, **kwargs    File
"/home/wizardcoder/1_wizardcoder-mindfommers/mindfommers/tra
iner/causal_language_modeling/caus al_language_modeling.py", line 104, in train

**kwargs) File "/home/wizardcoder/1_wiza rdcoder-mindfomers/mindfomers/t
rainer/base_trainer.py", line 631, in training_process   initial epoch=config.
runner config.initial epoch) File
*/home/xxx/miniconda3/envs/test08237Lib/python3 .7/site-packages/mindspore/t
rain/model.py", line 1066, in train    initial epoch=initial epoch)    File
"/home/xxx/miniconda3/envs/ test0823/lib/python3.7/site-
packages/mindspore/train/model .py", line 113, in wrapper    func(self. *args,
**kwargs) File */home/xxx/miniconda3/envs/t est0823/lib/python3.7/site-
packages/mindspore/train/model.py", line 620, in _train    cb params, sink size,
initial epoch, valid infos)  File
/home/xxx/miniconda3lenvs/test08237lib/python3.7/site-
packages/mindspore/train/model .py", line 709, in _train_da taset sink process list
callback .on_train step end(run context  File
"7home/xxx/miniconda3/envs/test0823/lib/python3.7/site-
packages/mindspore/train/callback/_cal lback.py", line 412, in on_train_step_end

cb.on train_step_end( run_ context)  File  /home/liyejun/miniconda3/envs/
test0823/lib/python3. 7/site-packages/mindspore/train/callback/_callback.py", line
254,. in on_train_step_end    self.step_end(run_context)  File
"/home/xxx7miniconda3/envs/test0823/lib/python3.7/site-
packages/mindspore/train/callback/_checkpoint.py". line 461, in step_end

self._save_ckpt(cb_params)  File "/home/wizardcoder/1_wizardcoder-
mindfomers/mindfomers/core/c allback/callback.py", line 481, in _save_ckpt

cb_params.train_ network.exec_checkpoint_graph()   File
"/home/xxx/miniconda37envs/test0823/Lib/python3 .7/site-
packages/mindspore/nn/cell.py". line 976. in exec_checkpoint_graph
cell_graph_executor(self, phase='save')   File
"/home/xxx/miniconda3/envs/test0823/lib/python3.7/site-
packages/mindspore/common/api -py", line 1672, in _call_    return self.run(obj,
*args. phase-phase)  File hae/xxxy/aincandas/ens/tert23/lib/pythens. 7/site
padkags/indspare/camon/apPr. uine 17nin nmreturn self._graph _executo r((), exe
phase)   File "/home/xxx/miniconda3/envs/test0823/lib/python3.7/site-
packages/mindspore/train/callback/_callback .py", line 88, in checkpo
int_cb_for_save_op  _fill_param_into_net(CUR NET, parameter list)  File
```

```
"/home/xxx/miniconda3/envs/test0823/lib/python3.7/site-
packages/mindspore/train/callback/_callback.py", line 68, in _fill_param_into_net

load_ param_into_ net(net, parameter dict, strict load=True) File
"7home/Tiyejun/miniconda3/envs/test0823/lib/python3.7/site-
packages/mindspore/train/serialization.py", line 1222, in load param_into_net

update_param (param, new_param, strict load)  FiTe "/home/xxx/miniconda3/envs/
test823/lib/python3.7/site-pack ages/mindspore/train/serialization .py", line 125,
in _update_param

RuntimeError: For "load param into net', accu_grads.backbone .blocks.5.attention.
projection.weight in the argument 'net' should have the same shape as accu_
grads.backbone .blocks.5.

ttention.projection.weight in the argument 'parameter dict'. But got its shape
(1536, 6144) in the argument 'net' and shape (3072, 6144) in  the argument
'parameter_dict '.May you

need to checkwhether the checkpoint you loaded is correct or the batch size and so
on in the 'net' and 'parameter dict' are same.
```

3. 根因分析

保存模型需要使用到 callback.py 中的_save_ckpt()功能，但是在使用 Ascend 上使用时，会走进如下代码的第一个 if 判断代码部分，从而导致报错。

```
self._last_triggered_step = cb_params.cur_step_num
if context.get_context("enable_ge") and os.getenv("MS_ENABLE_REF_MODE", "O") == "O":
    set_cur_net(cb_params.train_network)
    cb_params.train_network. exec_checkpoint_graph()
if "epoch_num" in self._append_dict:
    self._append_dict["epoch_num"] = self._append_epoch_num+ cb_params.cur_epoch_num
if "step_num" in self._append_dict:
    self._append_dict["step_num"] = seLf._append_step_num + cb_params.cur_epoch_num
* cb_params.batch_num
```

4. 解决方案

目前的解决方案有两个（目标就是不走进这个条件判断里面）

```
self._last_triggered_step = cb_params.cur_step_num
if context.get_context("enable_ge") and os.getenv("MS_ENABLE_REF_MODE", "O") == "O":
    set_cur_net(cb_params.train_network)
    cb_params.train_network. exec_checkpoint_graph()
```

1. 注释掉上面 if 判断的代码，问题可以解决。

2. 设置 export MS_ENABLE_REF_MODE=1，上述代码则不需要注释。注意：有些环境中设置 MS_ENABLE_REF_MODE=1 可能会报错，可能是 CANN 版本等原因，这需要在适合的环境中使用。

2.3 MindSpore 开启 profiler 功能报错 IndexError:list index out of range

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.1

执行模式： 不限

2. 报错信息

2.1 问题描述

使用 Ascend 环境，按照 Performance_Tuning.md 方法启动 profiler 时报错如下：

```
Traceback (most recent call last):  File "wizardcoder/run_wizardcoder.py", line 149,
in <module>    device_id=args .device_id)  File "wizardcoder/run_wizardcoder.py",
line 81, in main

1Ctask.train(train_checkpoint=ckpt,  resume=resume)    nile /home /vizardoderÄ,
vizardcder aindformers/aindformers/trainer/trainer.py",une   is_full
config=True,**kwargs)   File-"/home/wizardcoder/1_wizardcoder-
mindformers/mindformers/trainer/c ausal_language_modeling/caus
al_language_modeling.py", line 104, in train  **kwargs) File
"/home/wizardcoder/1_wizardcoder-mindformers/mindformers/t rainer/base_trainer.py",
line 631, in training_process   initial_epoch=config.runne r_config.initial_epoch)
File "/root/miniconda3/envs/wizardcoder/lib/python3.7/site-
packages/mindspore/train/model .py", line 1066, in train
initial_epoch=initial_epoch)    File
"/root/miniconda3/envs/wizardcoder/lib/python3.7/site-
packages/mindspore/train/model.py" , line 113, in wrapper    func(self, *args,
**kwargs) File "/root/miniconda3/envs/wizardcoder/lib/python3.7/site-
packages/mindspore/train/model.py" , line 613, in _train
self._train_process(epoch, train_dataset, list_callback, cb_params, initial_epoch,
valid infos)   File "/root/miniconda3/envs/wiz ardcoder/lib/python3.7/site-
packages/mindspore7train/model.py", line 921, in _train_process

list_callback.on_ train_step_end(run_context)  File
"7root/miniconda3/envs/wizardcoder/lib/python3.7/site-pac
kages/mindspore/train/callback/_callback .py", line 413, in on_train_step_end
cb.on_train_step_end(run_context)    File "/root/miniconda3/envs/wizardcode
r/lib/python3.7/site-pac kages /mindspore/train/callback/_callback .py", line 255,
in on_train_step_end  self.step_end(run_context  File
"/home/wizardcoder/1_wizardcoder -mindformers/mindformers/core/cal
lback/callback.py", line 593, in step_end self.profiler.analyse() File
"/root/miniconda3/envs/wizardcoder/lib/python3.7/site-
packages/mindspore/profiler/profiling.py", line 594, in analyse 710
self._analyse(offline_path=of fline_path) File "/root/miniconda3/envs/wizardcode
r/lib/python3.7/site-packages/mindspore/profiler/profiling.py", line 634, in
Qahalyse Fi1elf, rastemdnananda3/envs /wizardcoder/Lib/python3.7/site-packages
/mindspore/profiler/profiling.py", line 1021, in _ascend_analyse
self._ascend_graph_analyse()    File
"/root/miniconda3/envs/wizardcoder/lib/python3.7/site-
```

```
packages/mindspore/profiler/pro filing.py", line 1250, in _ascend graph_analyse

op_summary, op_statistic, steptrace = ascend_graph_msprof_analyse( source_path)
File "/root/miniconda3/envs/wizardcoder/Tib/python3.77site-packages
/mindspore/profiler/profiling .py", line 298, in _ascend_graph_msprof_analyse    df
op summary, df op statistic, df step trace = msprof analyser.parse()    File
"/root/mıniconda3/envs/wizardcoder/lıb/python3.7/site-
packages/mindspore/prof1ler/parser/ascend_msprof_generator.py", line 97, in parse
self._read_op_summary() File "/root/miniconda3/envs
/wizardcoder/lib/python3.7/site-packages/mindspore/profiler/parser/ascend_msp
rof_generator.py", line 121, in _read_op_summary    row = [row[index.get( 'index')]
for index in self.op_summary_name.values()] File
"/root/miniconda3/envs/wizardcoder/lib/python3 . 77site-
packages/mindspore/profiler/parser/ascend_mspro f_generator.py", line 121, in
<listcomp>  row = [row[index.get( 'index')] for index in self.
op_summary_name.values()]

IndexError: list index out of range
```

3. 根因分析

原因是 Ascend 环境启动 profiler 需要配置一些环境命令。

4. 解决方案

1. unset 上述环境变量；

2. 安装 hccl_parser：pip install {CANN 包址}/latest/tools/hccl_parser-0.1-py3-none-any.whl，使用 env 命令查看 CANN 包地址；

3. Yaml 文件中 profile 设为 True。

2.4 MindSpore 多机运行 Profiler 报错 ValueError: not enough values to unpack (expected 4, got 0)

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.1.1

执行模式： 不限

2. 报错信息

2.1 问题描述

使用 Ascend+MindSpore2.1.1 环境中，配置 2 机 16 卡全参训练，开启 profile 时可以启动训练，但是在使用 profile 收集性能数据时出现问题，报错如下：

```
Thu 21 Sep 2023 11:00:57 [INFO] [MSVP] [2328699] msprof_common . py: start
analyzing data in "/home/wizardcoder/ 1_wizardcoder-mindformers-
916/research/output/profile/rank_15/p rofiler/
PROF_000001_20230921085903388_FMDGQIRGMNNQRFFB/device_7"
```

```
Thu 21 Sep 2023 11:00:57 [INFO] [MSVP] [2328699] msprof_common .py: It may take few
minutes, please be patient  Thu 21 sep 2023 11:01:07 [INFo] [MsvP] [2328699]
msprof_common.py: Analysis data in "/home/wizardcode r/1_wizardcoder-mindformers -
916/research/output/profile/rank_15/p rofiler/PROF_0€   0001 20230921085903388
FMDGQIRGMNNQRFFB/device 7" finished   [WARNING] ME(1932377:28147 3783364672,
MainProcess): 2023-09-21- 11:04:05.598.182 [mindspore/pro filer/profiling.py:1102]
[Profiler] Can not found cube fops and vector fops data in the

summary  [WARNINGÍ ME (1932377:281473783364672, MainProcess):2023-09-21 -
11:04:05.994.536 [mindspore/prof iler/parser/memory_usage_parser.py:135] The memory
file does not exist! Please ignore th

warning  if you are running heterogeneous training.    [WARNING]
ME(1932377:281473783364672,MainProcess):2023-09-21-11:04:05.994.877
[mindspore/profiler/profiling.py:1134l The file </home/wizardcoder/1_wizardcoder -
mindformers-916/resear    ch/output/profile/rank_15/profi ler/memory_usage_15.pb>
not found "aceback (most recent call last):    File
"wizardcoder/run_wizardcoder.py", line 149, in <module>

device id=args.device id)   File "wizardcoder/run wizardcoder.py", line 81, in main

task.train(train checkpo int=ckpt,resume=resume)   File-
"/home/wizardcoder/1_wizardcoder-mindformers-916/mindformers/trainer/trainer.py" ,
line 423, in train    is full config=True    —   **kwargs) File "
/home/wizardcoder/1 wizardcoder-mindformers-916/mindformers/t
rainer/causal_language_modeling/causal_language_modeling.py", line 106, in train
**kwargs) File "/home/wizardcoder/1_wizardcoder-mindformers-916/mindformers/t
rainer/base_trainer.py", line 644, in training_process

initial epoch=config.runner_config.initial_epoch)  File
"/root/anaconda3/envs/wizardcoder/lib/python3.7/site-
packages/mindspore/train/model.py" , line 1066, in train

initial epoch=init ial epoch)    File "/root/anaconda3/envs/wizar
dcoder/lib/python3.7/site-packages/mindspore/train/model.py" , line 113, in wrapper
func(self. *args, **kwargs) File "/root/anaconda3/envs/wizardcode
r/Lib/python3.7/site-packages/mindspore/t rain/model.py", line 613, in _train  self.
train process(epoch, train dataset, list callback, cb params, initial epoch, valid
infos)    File "/root/anaconda3/envs /wizardcoder/lib/python3.7/site-
packages/mindspore/train/model.py", line 921, in _train_process    list callback.on
train_step_end(run context)   File "7root/anaconda3/envs
/wizardcoder/lib/python3.7/site-packages/mindspore/t rain/callback/_callback .py",
line 412, in on_train_step_end

cb.on train step end( run context)   File "/root/anaconda3/envs7wizardcode
r/lib/python3.7/site-packages /mindspore/train/callback/_callback .py", line 254,
in on_train_step_end  self.step_end(run_context)  File
"/home7wizardcoder/1_wizardcoder-mindformers -916/mindformers/core/cal
Lback/callback.py", line 630, in step_end    self.profiler.analyse() File "
"/root/anaconda3/envs/wizardcode r/lib/python3.7/site-
packages/mindspore/profiler/profiling.py", line 579, in analyse  self. ascend
analyse()

File Cself.ascend graph analyse()
/root/anacondas/enws/wizardeoder/Lib/pythona.7/site-
```

```
packages/mindspore/prefiter/profiling. py, Line 970, in ascend anatys File
"/root/anaconda3/envs /wizardcoder/lib/python3.7/site-packages
/mindspore/profiler/profiling.py", line 1205, in ascend graph analyse  self. ascend
graph hccl analyse(source path)    File
"/root/anaconda3/envs/wizardcoder/lib/python3.7/site-
packages/mindspore/profiler/profiling.py" . line 1153. in ascend graph hccl analyse

File "7root/anaconda3/envs/wizardcoder/lib/python3.7/s ite-
packages/mindspore/profiler/parser/ascend_hccl _generator.py", line 148, in parse
raw = self._iteration_analyse(hccl_detail data, iteration_ id)   File
"/root/anaconda3/envs/wizardcoder/lib/python3.7/site-
packages/mindspore/profiler/parser/as cend hccl_generator.py", line 222, in
_iteration_analyse  link info = self. link info analyse(hccl detail data)

File "7root/anaconda3/envs/wizardcoder/lib/python3.7/site-
packages/mindspore/profiler/parser/ascend_hccl_generator.py", line 247, in
_link_info_analyse    transport_information[ 'RDMA'] = self. rdma_analyse(groupby_t
ransport)

File "/root7anaconda3/envs/wizardcoder/Tib/python3.7/site-
packages/mindspore/profiler/parser/ascend _hcc l_generator.py", line 102, in
_rdma_analyse

thread_groups, _, _, _ = np.unique(groupby_transport['tid']) ValueError: not enough
values to unpack (expected 4, got 0)
```

3. 根因分析

原因应该和两机网络通信有关。

4. 解决方案

```
File "/root7anaconda3/envs/wizardcoder/Tib/python3.7/site-
packages/mindspore/profiler/parser/ascend _hcc l_generator.py", line 102, in
_rdma_analyse
   thread_groups, _, _, _ = np.unique(groupby_transport['tid']) ValueError: not
enough values to unpack (expected 4, got 0)
```

解决办法是把上面报错提示中的“_, _, _”去掉。

2.5 MindSpore 开启 summary 功能失效

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.1

执行模式： 不限

2. 报错信息

2.1 问题描述

在 yaml 文件的 callbacks 中配置 SummaryMonitor 后，训练时无法生成 summary 数据，如下为配置文件信息：

```
lcallbacks:
  - type: MFLossMonitor
  - type: SummaryMonitor
    collect_freq: 50
    keep_default_action: True
```

3. 根因分析

原因和 MindSpore Transformers 版本有关。

4. 解决方案

旧版本 MindSpore Transformers 的 base_trainer.py 文件中阴影部分的代码会导致 SummaryMonitor 失效，所以需要将其注释掉。最好的方法是将 mindformers 更新到最新版本。

```
# # del SummaryMonitor, or it will crash
# for index, callbacks inenumerate(self.config.callbacks):
#     if callbacks["type"] == "SummaryMonitor":
#         del self.config.callbacks[index]
```

2.6 MindSpore 模型权重功能无法保存更新后的权重

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.1.1

执行模式： 不限

2. 报错信息

2.1 问题描述

使用 CANN7.0+MindSpore2.1.1，训练 loss 收敛后保存模型，发现保存的不是训练之后的模型权重，而是初始权重，如下所示:

初始化权重信息：

```
Tensor (shape=[6144, 6144], dtype=Float32 , value=
[[ 1.00874314e-02 6. 56688539e-03 -1. 91919804e-02 … 5.64888678e-03 -2. 20796163e-
03 4.29333374e-03]
[-1.58290453e-02 -7. 58421593e-05 -4. 00598830e-04 -1.14668552e-02 - 1.13498308e-02
2.55679921e-03]
[ 5.50412433e-03 1. 71776358e-02 -2.33 142842e-02 1.49840652e-03 1.28344800e-02
8.48229043e-03]
[ 5.53949829e-03 -1. 25942379e-02 6.60581887e-03    -8.29280645e-04 -1.08878603e-02
-4.47568891e-04]
[ 1.70315057e-03 3. 48439417e-03 7.81778805e-03  C -7.27706531e-04  7.55816605e-03
-2.24832026e-03]
[ 8.41980707e-03 6. 66081626e-03 -4. 32111416e-03 -1.40487147e-03 - 4.82343137e-03
-4. 50384151e-03]])
```

训练过程中打印的权重信息：

```
Tensor(shape=[6144, 6144], dtype=Float32, value=
[[ 1.01261148e-02 6. 58480451e-03 -1. 90741140e-02     5.70917921e-03 -2.
18650815e-03   4.28082701e-03]
[-1.57715511e-02 -4. 06106119e-05 -3.- 39789083e-04    -1.13997944e-02 1.13531193e-
02 2.33310624e-03]
[ 5.52728493e-03 1.71936452e-02 -2. 32268739e-02   1.55355304e-03 - 1.29088536e-02
8.45980365e-03]
[ 5.42893354e-03 -1. 26162879e-02 6.61681918e-03  -8.39053362e-04 -1.09538278e-02 -
4.17841686e-04]   1.68025843e-03 3.56728001e-03 7.80858565e-03 .. -8.41747446e-04
7.55744614e-03 -2.15848 139e-03]
[ 8.49595107e-03 6.70013577e-03 -4.35717218e-03   -1.36322062e-03 4.75577544e-03 -4.
33780858e-03]])
```

模型保存的权重信息：

```
-callbacks begin-
Param: backbone.blocks.0. attention.densel.weight with shape (6144, 6144)
[[ 1.0087431e-02 6.5668854e-03 -1.9 191980e-02  - 5.6488868e-03
-2.2079616e-03 4.2933337e-03]
[-1.5829045e-02 -7 .5842159e-05 -4.0059883e-04 ... -1.1466855e-02
1.1349831e-02 2.5567992e-03]
[ 5.5041243e-03 1. 7177636e-02 -2.3314284e-02  1.4984065e-03
1.2834480e-02 8.4822904e-03]
[ 5.5394983e-03 -1.2594238e-02 6.6058189e-03 ..... -8 .2928065e-04
-1.0887860e-02 -4. 4756889e-04]
[ 1.7031506e-03 3. 4843942e-03 7.8177880e-03 ... -7.. 2770653e-04
7.5581660e-03 -2.. 2483203e-03]
[ 8.4198071e-03 6. 6608163e-03 -4.3211142e-03  .-1.4048715e-03
4.8234314e-03 -4.5038415e-03]]
-callbacks end
```

3. 根因分析

根据上述信息发现最终保存的模型权重还是和初始化权重一样，表示并没有保存到训练之后的权重。

4. 解决方案

升级版本到 MindSpore 2.2 之后，模型保存可以正常工作。

2.7 MindSpore 开启 MS_DISABLE_REF_MODE 导致报错 The device address type is wrong:type name in address:CPU,type name in context:Ascend

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 报错信息

2.1 问题描述

使用 MindSpore2.2+CANN 7.0 环境，并行策略为 dp:mp:pp=1:4:2 时可正常跑通训练，但是改变并行策略为 dp:mp:pp = 1:8:1 时出现如下报错：

```
2023-10-28-10:52:46.336.155 -> 2023-10-28-10:52:46.336.477
For more details, please refer to the FAQ at
https://www.mindspore.cn/docs/en/master/faag/data processing.htmlL.
Traceback (most recent call last):
File "wizardcoder/run_wizardcoder.py", line 170, in <module>
merge_file=args.merge_file)
File "wizardcoder/run_wizardcoder.py", line 104, in main
task.finetune(finetune_checkpoint=config.load_checkpoint,
auto_trans_ckpt=config.auto_trans_ckpt, resume=resume)
File "/home/wizardcoder/2_wizardcoder-mindformers -
1019/mindformers/trainer/trainer.py", line 462, in finetune
is_full_config=True, **kwargs)
File "/home/wzardcoder/2_wizardcoder-mindformers -
1019/mindformers/trainer/causal_language_modeling/causal_language_modeling.py",
Line 163, in train **kwargs)
File "/home/wizardcoder/2 wizardcoder-mindformers -1019/mindformers/trainer/base
trainer.py", Line 653, in training process
initial_epoch=config.runner_config.initial_epoch)
File ""/home/miniconda3/lib/python3.7/site-packages/mindspore/train/model.py", Line
1073, in train
initial_epoch=initial_epoch)
File "/home/miniconda3/lib/python3.7/site-packages/mindspore/train/model.py", line
114, in wrapper
func(self, *args, **kwargs)
File "/home/miniconda3/1ib/python3.7/site-packages/mindspore/train/model.py", line
624, in _train
cb_params, sink size, initial_epoch, valid infos)
File "/home/miniconda3/1lib/python3.7/site-packages/mindspore/train/model.py", line
708, in _train_dataset_sink_process
outputs = train network(*inputs)
File "/home/miniconda3/lib/python3.7/site-packages/mindspore/nn/cell.py", line 680,
in _call
out = self.compile and run(*args, **kwargs)
File "/home/miniconda3/lib/python3.7/site-packages/mindspore/nn/cell.py", line 1023,
in compile _and_run
return _cell_graph_executor(self, *new_args, phase=self.phase)
File "/home/miniconda3/lib/python3.7/site-packages/mindspore/common/api.py", line
1589, in __call
return self.run(obj, *args, phase=phase)
File "/home/miniconda3/lib/python3.7/site-packages/mindspore/common/api.py", line
1628, in run
return self. exec_pip(obj, *args, phase=phase real)
File "/home/miniconda3/lib/python3.7/site-packages/mindspore/common/api.py", line
121, in wrapper
results = fn(*arg, **kwargs)
File "/home/miniconda3/lib/python3.7/site-packages/mindspore/common/api.py", line
1608, in _exec_pip
return self. graph executor(args, phase)
```

```
RuntimeError: The device address type is wrong: type name in address:CPU, type name
in context:Ascend
```

3. 根因分析

分析发现是开启“MS_DISABLE_REF_MODE”环境变量所导致。

4. 解决方案

unset 此环境变量后即可正常跑通。

2.8 MindSpore 和 tbe 环境相关报错

1. 根因分析

大多数是因为当前虚拟环境的 te 版本和 MindSpore、CANN 包不匹配。

2. 解决方案

尝试卸载相关 whl 包，重装对应 CANN 路径下的 whl 包。

```
pip uninstall te hccl auto_tune schedule_search -y
pip install {CANN 安装路径}/latest/aarch64-linux/lib64/*.whl
```

2.9 Ascend 上构建 MindSpore 报头文件缺失

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.1.1

执行模式： 不限

2. 报错信息

2.1 问题描述

Ascend 上构建 MindSpore 报头文件缺失,报错如下：

```
In file included from
/home/dev/mindspore/mindspore/ccsrc/transform/graph_ir/op_declare/experiment_ops_de
clare.h:21:0,
from /home/dev/mindspo re/mindspore/ccsrc/transform/graph_
ir/op_declare/experiment_ops_declare.cc:17:/home/dev/mindspo
re/mindspore/ccsrc/transform/graph_ir/op_declare/experiment_ops_decla re.cc: In
lambda function:
/home/dev/mindspore/mindspore/ccsrc/transform/graph_ir/op_declare/op_declare_macro.
h:191:18: error: 'using element_type = class ge: : op: : FlashAttentionScore (aka
class ge::op::F  lashAttentionScore}' hasno member named 'set_attr_input_layout';
did you mean 'set_attr_scale_value'? (void)p->set_attr_##name(ConvertAny(value, _VA
ARGS)):
/home/dev/mindspore/mindspore/ccsrc/transform/graph_ir/op_declare/experiment_
ops_declare.cc:33:20: note: in expansion of macro 'ATTR_DESC'   {"input_layout",
```

```
ATTR_DESC( input_layout, AnyTraits<std: :string>())},
/home/dev/mindspore/mindspore/ccs
rc/transform/graph_ir/op_declare/experiment_ops_declare.cc: In lambda function:
/home/dev/mindspore/mindspore/ccsrc/trans form/graph ir/op_declare/op_declare
macro.h:196:32: error: 'using element_type = class ge: :op: :FlashAttentionScore
{aka class ge::op::F   lashAttentionScore}' has no member named
'get_attr_input_layout'; did you mean 'get_attr_scale_value'?   auto real_value =
p->get_att r_##name();
/home/dev/mindspore/mindspore/cc src/transform/graph_ ir/op declare/experiment ops
declare .cc:33:20: note: in expansion of maCrO 'ATTR DESC'   {"input_layout",
ATTR_DESC( input_layout, AnyTraits<std: :string>())},    chenqxia
/home/dev/mindspore/mindspore/ccs
rc/transform/graph_ir/op_declare/experiment_ops_declare.cc: In lambda function:
/home/dev/mindspo
re/mindspore/ccsrc/transform/graph_ir/op_declare/op_declare_macro.h:191:18: error:
'using element_type = class ge::op::FlashAttentionScoreGrad {aka class ge::0
p: : FlashAttentionScoreGrad}' hasno member named 'set_attr_input_layout'; did you
mean 'set_att r_scale_value'? (void)p->set_attr_##name(ConvertAny(value,—VA_ARGS ):
/home/dev/mindspore/mindspore/cc
src/transform/graph_ir/op_declare/experiment_ops_declare .cc:52:20: note: in
expansion of TmacrO 'ATTR_DESC'
{"input_layout", ATTR_DESC( input_layout, AnyTraits<std: :string>())},
/home/dev/mindspore/mindspore/ccs
rc/transform/graph_ir/op_declare/experiment_ops_declare.cc: In lambda function:
/home/dev/mindspore/mindspore/ccsrc/transform/graph_ir/op_declare/op_declare_macro.
h:196:32: error: 'using element_type = class ge: : op: : FlashAttentionScoreGrad
{aka class ge::0  p:: FlashAttentionScoreGrad}' has no member named 'get_att
r_input_layout'; did you mean 'get_attr_scale_value'?  auto real_value = p->get att
r_##name():
/home/dev/mindspore/mindspore/ccs
rc/transform/graph_ir/op_declare/experiment_ops_declare.cc :52:20: note: in
expansion of macro *ATTR_DESC'
{"input_layout", ATTR_DESC( input_layout, AnyTraits<std: :string())},
/home/dev/mindspore/mindspore/ccs
rc/transform/graph_ir/op_declare/experiment_ops_declare.cc: In lambda function:
/home/dev/mindspo re/mindspore/ccsrc/transform/graph
ir/op_declare/op_declare_macro.h:223:18: error: 'using element_type = class
ge::op::FlashAttentionScoreGrad {aka class ge::0
p::FlashAttentionScoreG rad}' has no member named 'update output desc dpse' i did
you mean 'update_output_desc_dq'?
(void)p->update_output_desc_##name(desc); -henpxao
```

3. 根因分析

MindSpore 和 CANN 包版本不匹配。

4. 解决方案

需要将对应 CANN 包的 latest/opp/built-in/op_proto/inc/experiment_ops.h 文件复制并覆盖 mindspore 目录下的 graphengine/910b/third_party/fwkacllib/inc/ops/experiment_ops.h，不需要清缓存，直接重新执行构建命令即可。

2.10 MindSpore 开启 profile，使用并行策略报错 ValueError: When dıstrıbuted loads are slıced we1ghts, sınk mode must be set True.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.1

执行模式： 不限

2. 报错信息

2.1 问题描述

使用离线切分方法切好模型后，加载切分后的模型进行训练，出现以下报错：

```
Traceback (most recent call last):  File "wizardcoder/ run_wizardcoder.py", line
148, in <module>    device_id=args.device_id)  File "wizardcoder/
run_wizardcoder.py", line 90, in main    task.finetune(finetune_checkpoint=config.
load_checkpoint, auto_trans_ckpt=config.auto_trans_ckpt, resume=resume)    File
"/home/wizardcoder/1_wizardcoder-mindformers-916/mindformers/trainer/trainer.py",
Tine 522, in finetune    is full config=True. **kwargs)  File "/home/wizardcoder/1
wizardcoder-mindformers-916/mindforme rs/trainer/caus al_language_modeling/caus
al_language_modeling.py", line 106, in train    **kwargs) File "/home/wizardcode
r/1_wizardcoder-mindformers -916/mindformers/trainer/base_t rainer.py", line 616,
in training_ process transform and load checkpoint(c onfig, model, network, dataset)
File "/home/wizardcoder/1 wizardcodermindformers -
916/mindformers/trainer/utils .py", line 300, in transform and_load checkpoint
build model(config, model. datas et, do eval=do eval, do predict-do predict)  File
"/home/wizardcoder/1 wizardcoder-mindformers-916/mindformers/trainer/utils.py".
line 317, in build model  raise ValueError( "when distributed loads are s liced
weights. sink mode must be set True

ValueError: When dıstrıbuted loads are slıced we1ghts, sınk mode must be set True.
```

3. 根因分析

分析配置文件可知，此时开启了 profile 功能，当前不支持两者同时使用。

4. 解决方案

不同时使用这两个功能。

2.11 模型启动时报 Malloc device memory failed

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.1

执行模式： 不限

2. 报错信息

程序刚启动报 Malloc device memory failed。

```
2023-08-24 09:03:49,594 - mindformers - INFO - Network Parameters: 13264 M.
Traceback (most recent call last):
   File "/home/ma-user/work/mindformers/research/baichuan/run_baichuan_13b.py",
line 115, in <module>
      main(task-args. task,
   File "/home/ma-user/work/mindformers/research/baichuan/run_baichuan_13b.py",
line 82, in main
      result - trainer.predict(input_data-predict_data,
   File "/home/ma-user/anaconda:s/envs/MindSpore/1ib/python3.9/site-
packages/mindformers/trainer/trainer.py", line 678, in predict
      output result = self.trainer predict(
   File "/home/ma-user/anaconda:3/envs/MindSpore/1ib/python3.9/site-
packages/mindformers/trainer/causal_language_modeling/causal-language_modeli   ng-
py", line 309, in predict
      return self.predict_process(config-config,
   File "/home/ma-user/anaconda3/envs/MindSpore/1ib/python3.9/site-
packages/mindformers/trainer/base_trainer.py", line 744,. in predict_process
      transform_and_load_checkpoint(config, model, network, None, do predict True)
   File "/home/ma-user/anaconda:8/envs/MindSpore/1ib/python3.9/site-
packages/mindformers/trainer/utils.py", 1ine 294, in transform_and_load_checkpoint
      raise FileNotFoundError(f"The load checkpoint must be correct,
FileNotFoundError: The load_checkpoint must be correct, but get /home/ma-
user/work/mindformers/research/ckpt_ dir/tıansform_ 13b
[ERROR] PIPELINE(38220, ffff9632db30,python):2023-08-24-09:03:49.648.021
[mindspore/ccsrc/pipeline/jit/pipeline.cc:2073] ClearResAtexit] Check exception
before process exit: Ascend kernel runtime initialization failed. The details refer
to "Ascend Error Message..
Framework Error Message:
Malloc device memory failed, free memory size is less than half of total memory
size.Device 0 Device HBM total size:34359738368 Device HBM free size:2094153728
maybe other processes occupying this card, check as: ps -eflgrep python
C++ Call Stack: (For framework developers)
mindspore/ccsrc/plugin/device/ascend/hal/device/ascend_kernel_runtime .cc:413 Init
mindspore/ccsrc/plugin/device/ascend/hal/device/ascend_memory_adapter.cc:63
Initialize
```

3. 解决方案

这种情况是因为本次运行的卡显存已被占用超过一半，可以通过 npu-smi info 查看占用卡的进程号，再通过 ps -aux | grep <pid_num> 查看该进程具体信息，如果要终止进程可以通过 kill -9 <pid_num>, 如果不能终止进程，可以通过指定其他空闲 device_id 重新运行程序。

2.12 MindSpore 盘古模型报错 Failed to allocate memory.Possible Cause: Available memory is insufficient.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 静态图

2. 报错信息

2.1 问题描述

MindSpore 盘古模型报错 Failed to allocate memory.Possible Cause: Available memory is insufficient.

2.2 报错信息

```
- Ascend Error Message:
-----------------------------------------------------
EL0004: Failed to allocate memory.
       Possible Cause: Available memory is insufficient.
```

或者是

```
The Free Device Memory Size is 28.2318 GB, variable
memory_max_size/max_device_memory should be in range (0-28909.4]MB, but got 30720MB,
please set the context key ‘variable memory_max_size’/‘max_device_memory’ in valid
range.
```

3. 根因分析

1、推动云平台升级升级 driver 至 23.0rc3 以上（记得各种包配套问题），可降低 hccl 占用的显存。

2、主函数 python 修改 context，或修改 mindformers 中 yaml 配置文件的 max_device_memory 配置，尝试调小分配给 MindSpore 的显存，留更多显存给 HCCL/Driver。

```
context.set_context(max_device_memory="28G") # 910 显存设置，B4 可尝试 26G 等
context.set_context(max_device_memory="57G") # 910 显存设置，B3 可尝试 57G 等
```

3、启动脚本中添加以下环境变量。

```
echo "---------------npu-smi info-----------"
npu-smi info
export MS_DEV_CELL_REUSE=1   #开启 cell 共享
```

或只增加该两项环境变量，pangu-sigma 38b 尝试可行。

```
export MS_ENABLE_FORMAT_MODE=1
export MS_MEMORY_STATISTIC=1
```

如果所有的方法都没用，不断降低 bs，和网络 layers，和 max_device_memory，发现最终报错的算子在 hccl 侧显存不够。

```
E19999: Inner Error!
E19999  Call ops_kernel_info_store LoadTask fai1[FUNC: Distribute][FILE:
hcc1_task_info. cc][LINE: 320]
          TraceBack (most recent call last):
          Call ops kernel_info store unloadltask fai1LFUNC:
ReleaseJLFILE:hccl_task_info. ccl[LINE: 657|
          GraphManager RunGrapWithStrteamhAsync failed, session id= 0, graph id =
286, stream= Oxaaaae871edd0. [FUNC: RunGraphWithStreamAsync][FILE:inner_session.
```

```
cc][LINE: 516]
          [Run][Graph]Run graph with stream asyn failed, error code -1343225860,
session id = 0,graph id = 286, stream = Oxaaaae871edd0. [FUNC:
RunGraphWithStreamAsync][FILE: ge_api. cc][LINE: 774]
(Please search "Ascend Error Message" at https://www. mindspore.cn for error code
description)
```

2.13 Ascend910 环境分离部署时请求超时

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend910

软件环境: MindSpore 版本: 2.2.10

执行模式： 不限

Python 版本: 3.8.15

操作系统平台: linux

2. 报错信息

2.1 问题描述

在 Ascend910 环境，进行单节点 8 卡分离部署，全量模型正常执行，但是增量模型请求获取 KV 数据时，GE 报错请求超时。

3. 根因分析

通过日志分析，怀疑是节点内 device 通信问题，使用的是 Ascend910 环境。

4. 解决方案

numa_config_global.json 内容修改如下所示：

```
"nodes_topology":
{
   "protocol": "TCP",
   "type" : "star",
   "topos" : [
      {
         "plane id" : 0,
         "devices" : [0, 4]
      },
      {
         "plane id" : 1,
         "devices": [1, 5]
      },
      {
         "plane id" : 2,
         "devices": [2, 6]
      },
      {
         "plane id" : 3,
         "devices": [3, 7]
```

```
        }
    ]
```

修改为：

```
"nodes_topology":
{
    "protocol": "RDMA",
    "type" : "star",
    "topos" : [
        {
            "plane id" : 0,
            "devices" : [0, 4]
        },
        {
            "plane id" : 1,
            "devices": [1, 5]
        },
        {
            "plane id" : 2,
            "devices": [2, 6]
        },
        {
            "plane id" : 3,
            "devices": [3, 7]
        }
    ]
```

3 配置问题案例

3.1 MindSpore 报错 ImportError cannot import name "build dataset loader' from 'mindformers.dataset. dataloader'

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.1

执行模式： 不限

2. 报错信息

2.1 问题描述

运行如下脚本后出现数据处理的报错：

```
python run_mindformer.py --config configs/gpt_bigcode/run_gpt_bigcode. yaml --
run_mode train \
                    --device_target Ascend \
                    --train dataset dirdata/qpt biqcode
```

2.2 报错信息

运行如下脚本后出现数据处理的报错：

```
ImportError cannot import name "build dataset loader' from 'mindformers.dataset.
dataloader' (/opt/mindformers/mindformers/dataset/dataloader/init.py)
```

3. 根因分析

报错表示无法加载 build_dataset_loader 函数。从报错信息给定的位置查找发现 __init__.py 文件中没有添加这个函数。

4. 解决方案

只需在__init__.py 文件中添加 build_dataset_loader 函数即可。

```
__all__ = [ 'Flickr8kDataLoader','Cifar100DataLoader',
'WMT16DataLoader','CLUENERDataLoader', 'SQuADDataLoader',
'ADGenDataLoader','MultiImgCapDataLoader','build dataset loader']
```

3.2 MindSpore 大模型报错 xxx is not a supported default model or a valid path to checkpoint.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.0

执行模式： 不限

2. 报错信息

xxx is not a supported default model or a valid path to checkpoint.

```
Traceback (most recent call last):
   File "/hone/ma user/work/aindformers-
rc88/aindformers/nindformers/tools/register/register.py", 1ine 192,
inget_instance_fron_cfg          return obj_cls(*angs)
   File "/hone/ma-user/work/aindforwers-
rc08/alndforwers/alndformers/versloncontrol.py", 1ine 49, in decorator
       func(*args, *loeangs)
   File "/hone/ma-user/work/alndforwers-
rc0a/aindformers/elndformers/models/1lana/llaa.py", line 311, in _init_
       self.load_checkpaint(config)
   File "/home/ma-user/work/aindforeers-
rc08/aindformers/alndformers/models/bse_model.py", 1ine 84, in load_checkpoint
       ralse ValueErnorf"lcheckpoint name or oath) is nota suooorted default nodel"
ValueError: /data/qwen/folder1/1lama2_13b/1lama2_7b.ckpt 1s nota supported default
model or a validpath to checkpoint, please select fron['1lama_7b','11ama_13b,
'11ama 65b','11ama2_7b','1lama2 136',  11ama2_70b','llasa 7b lora'].
```

3. 解决方案

1.如果传入的是模型名称，确认传入的模型名称是否为问题提示中的模型名称。

2.如果传入的是模型路径，确认传入的模型路径所指向文件是否存在。

3.3 llama2 模型转换报错 ImportError: cannot import name 'swap_cache' from 'mindspore._c_expression'

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.0

执行模式： 不限

2. 报错信息

dev 分支出现 ImportError: cannot import name 'swap_cache' from 'mindspore._c_expression'。

在 pull dev 分支的 mindspore 中，在准备微调 llama2 的前期模型转换工作出现了如下的 bug。

```
Traceback (most recent call last):
   File "/home/ma-user/work/mindformers/scripts/mf_parallel0/run_mindformer.py",
line 19, in <module>
      from mindformers.tools.register import MindFormerConfig,ActionDict
   File "/home/ma-user/work/mindformers/scripts/mf_paral
lel0/mindformers/_init_.py" , line 34, in <module>
      from mindformers import model_runner
   File "/home/ma-
user/work/mindformers/scripts/mf_paralle10/mindformers/model_runner. py", line 25,
in <module>
   from mindspore. c expression import swap cache
ImportError: cannot import name 'swap cache' from "mindspore. c expression'
(/home/ma-user/anaconda3/envs/MindSpore)
```

3. 解决方案

该问题由使用分支与 mindspore 配套不匹配引起，dev 分支目前配套 mindspore 版本为 2.3.rc2，可以通过 mindspore 官网指引安装基础配套软件和 mindspore。

3.4 MindSpore 报错报错 BrokenPipeError: [Errno 32] Broken pipe

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.10

执行模式： Graph

2. 报错信息

```
Process ForkServerPoolWorker-6:
Traceback (most recent call last):
   File "/root/miniconda3/envs/ms21_py38/lib/python3.8/multiprocessing/pool.py",
line 131, in worker
      put((job, i, result))
   File "/root/miniconda3/envs/ms21_py38/lib/python3.8/multiprocessing/queues.py",
line368, in put
      self._writer.send_bytes(obj)
   File
"/root/miniconda3/envs/ms21_py38/lib/python3.8/multiprocessing/connection.py", line
200, in send bytes
      self. send bytes(mloffset:of fset + size])
   File
"/root/miniconda3/envs/ms21_py38/lib/python3.8/multiprocessing/connection.py", line
411, in _send bytes
      self. send(header + buf)
   File
"/root/miniconda3/envs/ms21_py38/lib/python3.8/multiprocessing/connection.py", line
368, in _send
      n = write(self. handle, buf)
BrokenPipeError: [Errno32] Broken pipe
```

3. 根因分析

报错信息为读写失败，定位方向为：查看代码中，是否有往磁盘读写文件的操作原因：

```
context.set_context(mode=context.GRAPH_MODE, device_target=opt.device_target,
enable_compile_cache=True, compile_cache_path="./cache", max_call_depth=4096)
```

代码中开启了编译缓存，但写入的路径，本地没有。

4. 解决方案

修改 context 为

```
context.set_context(mode=context.GRAPH_MODE, device_target=opt.device_target,
max_call_depth=4096)
```

删除 enable_compile_cache=True, compile_cache_path="./cache"。

3.5 昇腾 910 上 CodeLlama 报错 Session options is not equal in diff config infos when models' weights are shared, last session options

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend910

软件环境: MindSpore 版本: 2.2.10

执行模式： 不限

Python 版本: 3.8.15

操作系统平台: linux

2. 报错信息

2.1 报错信息

```
Session options is not equal in diff config infos when models' weights are shared,
last session options
```

3. 根因分析

根据日志提示，注释符号错误，不能使用分号";"

4. 解决方案

检查配置.ini 文件，注释配置不能用；

```
;[graph_kernel_param]
;opt_level=2
;disable_cluster_ops=MatMul,Reshape
```

注释配置应使用#

```
#[graph_kernel_param]
#opt_level=2
#disable_cluster_ops=MatMul,Reshape
```

3.6 集成通信库初始化 init()报错要求使用 mpirun 启动多卡训练

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.10

执行模式： 不限

Python 版本: 3.8.15

操作系统平台: linux

2. 报错信息

2.1 报错信息

```
Traceback (most recent call last):
  File "train.py", line 185, in <module>
    main()
  File "train.py", line 141, in main
    set_parameter()
  File "train.py", line 65, in set_parameter
    init()
  File "/home/ma-user/anaconda3/envs/MindSpore/lib/python3.7/site-
packages/mindspore/communication/management.py", line 172. in init
    init_cluster()
```

```
RuntimeError: Maybe you are trying to call 'mindspore.communication.init()' without
using 'mpirun', which will make MindSpore load several environment variables and
check their validation. Please use 'mpirun' to launch this process to fix this
issue, or refer to this link if you want to run distributed training without using
'mpirun': https://www.mindspore.cn/tutorials/experts/zh-
CN/master/paralle1/train_gpu.html
```

2.2 相关源码

```
if backend_name == "hccl":
       if _is_ps_mode():
           # Use MindSpore cluster to build network for Parameter Server traning.
           init_cluster()
           if _is_role_sched() or _is_role_pserver():
              raise RuntimeError("Parameter server and scheduler should use 'CPU' as
backend instead of 'Ascend'")
           if _get_ps_context("worker_num") == 1:
              GlobalComm.INITED = True
              _set_elegant_exit_handle()
              return
       if device_target != "Ascend":
           raise RuntimeError("For 'init', the argument  'backend_name' should be
'Ascend' to init hccl, "
                              "but got {}".format(device_target))
       if not host_init:
           _check_parallel_envs()
       GlobalComm.BACKEND = Backend("hccl")
       init_hccl()
       GlobalComm.WORLD_COMM_GROUP = HCCL_WORLD_COMM_GROUP
   elif backend_name == "nccl":
       init_cluster()
       GlobalComm.WORLD_COMM_GROUP = NCCL_WORLD_COMM_GROUP
   elif backend_name == "mccl":
       init_cluster()
       GlobalComm.WORLD_COMM_GROUP = MCCL_WORLD_COMM_GROUP
   else:
       raise RuntimeError("For 'init', the argument 'backend_name' must be nccl
while 'device_target' is GPU, "
                          "but got the 'backend_name' : hccl.")
```

3. 根因分析

这是因为昇腾机器，调用 init()来初始化集成通信库，没有显式指定初始化的集成通信库类型。报错日志的调用栈中走到了 init_cluster()，但是阅读源码发现，初始化昇腾的集成通信库 hccl（非 ps 模式）的时候并不会调用 init_cluster()，代码走到了 gpu 的集成通信库 nccl 的初始化分支。

4. 解决方案

调用 init 初始化集成通信库时显式指定后端，如在昇腾机器上调用 init("hccl")。

3.7 INFNAN 模式溢出问题

1. 两种模式的区别

饱和模式（不开 infnan 模式）：检查训练中间比较、计算操作的溢出情况，只要中间结果有溢出，就会丢弃当前 step，重新开始训练。非饱和模式（infnan 模式）：只检查梯度数据是否存在 INF/NAN，不检查中间结果。中间结果溢出，只要梯度正常，整网依然不会溢出，可以正常训练。

2. 开启 INFNAN 模式后的溢出问题

开启 infnan 模式之后网络可能出现持续溢出，可以尝试关闭 matmul 算子的 fixpipe 融合，方法如下：

以 CANN 7.0 为例，修改

```
/usr/local/Ascend/CANN-7.0/opp/built-in/fusion_pass/config/fusion_config.json
"UBFusion":{
    "TbeMatmulFixPipeFus ionPass":"on",
    "TbeConvDequantSigmoidMulAddFus ionPass": "off",
    "Conv2dTransDataFusionPass" : "off",
    "AutoSingleNormFusionPass": "off",
    "AutoSingleReduceFusionPass": "off"
}
```

其中"TbeMatmulFixPipeFus ionPass"这个值改成"off"。

3. 开启 INFNAN 输出为 Nan，关闭 INFNAN 输出正常

现象：INFNAN 模式下训练 loss 为 nan,910A loss 正常，经过 dump 比对发现在 attention 中 query 和 key 进行矩阵乘获取 score 时，计算结果超出了 fp16 的表示范围，在 INFNAN 模式下依然为 inf, 然而在 910A 上会将超出 fp16 表示范围的置为 65504。

目前可以使用以下几种方法：1，全局使用 fp32 训练 2，溢出部分手动改成 fp32 计算，3，全局修改成 bf16 训练。

4 数据处理案例

4.1 Mindrecoder 格式转换报错 ValueError: For 'Mul'. x.shape and y.shape need to broadcast. The value of x. shapel-2] or y.shape[-2] must be 1 or -1 when they a are not the same, but got x.shape = [2, 2047,2047] and y.shape = [1. 2048, 2048]

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 报错信息

2.1 问题描述

Mindrecoder 格式转换时报错：

```
For more details, please refer to the FAQ at https://www.mindspore.cn/docs
/en/master/fag/data processing.html
raceback (most recentcall last): File "wizardcoder/run_wizardcoder.py", line 149,
in <module> device_id=args.device_id)  File "wizardcoder/run_wizardcoder.py", line
81, in main
task.train(train_checkpoint=ckpt, resume=resume)
File "/home/wizardcoder/1_wiza rdcoder-mindformers-
916/mindformers/trainer/trainer.py", line 423, in train    is_full_config=True,
**kwargs)  File"/home/wizardcoder/1_wizardcoder-mindformers -916/mindformers/t
rainer/causal_language_modeling/causal_language_modeling.py", line 106, in train
**kwargs)
File "/home/wizardcoder/1_wizardcoder-mindformers-
916/mindformers/trainer/base_trainer.py", line 644, in training_process
initial_epoch=config.runne r_config.initial_epoch) File
"/root/anaconda3/envs/wizardcoder/lib/python3.7/site-packages
/mindspore/train/model.py", line 1066, in train
initial_epoch=initial_epoch)    File "/root/anaconda3/envs
/wizardcoder/lib/python3.7/site-packages/mindspore/t rain/model.py", line 113, in
wrapper func(self, *args, **kwargs) File
"/root/anaconda3/envsJwizardcoder/lib/python3.7/site-
packages/mindspore/train/model.py" , line 620, in _train
cb_params, sink_size, initial_epoch, valid_infos)  File
"/root/anaconda3/envs/wizardcoder/lib/python3.7/site-packages
/mindspore/train/model.py", line 703, in _train_dataset_sink_process  outputs =
train_network(*inputs) cner File "/root/anaconda3/envs/wizardcoder/l
ib/python3.7/site-packages /mindspore/nn/cell.py", line 637, in _call_ out =
self.compile_and_run( *args, **kwargs)   File
"/root/anaconda3/envs/wizardcoder/lib/python3.7/site-packages /mindspo
re/nn/cell.py", line 961, in compile_and_run  self.compile(*args, **kwargs)   File
"/root/anaconda3/envs/wizardcoder/lib/python3.7/site-packages/mindspore/nn/cell.py",
line 939, in compile    jit_config_dict=self._jit_config_dict, *compile_args,
**kwargs)
File "/root/anaconda3/envs/wizardcoder/lib/python3.7/site-pack
```

```
ages/mindspore/common/api .py", line 1623, in compile    result =
self._graph_executor.compile(obj, args, kwargs, phase, self._use_vm_mode())    File
"/root/anaconda3/envs/wizardcoder/lib/python3.7/site-
packages/mindspore/ops/primitive.py", line 647, in infer_    outltrack] =
fn(*(xltrack] for x in args))    File
"/root/anaconda3/envs/wizardcoder/lib/python3.7/site-
packages/mindspore/ops/operations/math_ops .py", line 81, in infer_shape
return get_broadcast_shape(x_shape, y_shape, self.name) File
"/root/anaconda3/envs/wizardcode r/lib/python3.7/site-
packages/mindspore/ops/_utils/utils .py", line 70, in get_broadcast_shape
raise ValueError(f"For " {prim name}', {arg_name}.shape and {arg_name2}.shape need
to”
ValueError: For 'Mul'. x.shape and y.shape need to broadcast. The value of x.
shapel-2] or y.shape[-2] must be 1 or -1 when they a are not the same, but got
x.shape = [2, 2047,2047] and y.shape = [1. 2048, 2048]
```

3. 根因分析

MindFormers 基于 MindSpore 语言，先将数据 tokenizer 化后再转换为 Mindrecoder 格式，使用 Mindrecoder 格式的数据来训练模型；transformers 则不需要这种特定的数据格式。

4. 解决方案

因为 mindformers 设计的自身特点，在 tokenizer 时候需要将切分数据长度设置为 seq_length + 1，之后再保存为 mindrecoder 格式，就不会报错。

4.2 数据处理成 Mindrecord 数据时出现 ImportError: cannot import name 'tik' from 'te'

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 问题描述

qwen1.5-72b 在 Ascend 环境下，数据集转换为 mindrecode 时出现 导入 tik 异常，qwen1.5-72b 全参微调时，进行数据处理成 Mindrecord 数据生成时，出现报错。

```
ImportError: cannot import name 'tik' from
'te'(/root/miniconda3/envs/qwen72b/lib/python3.9/site-packages/te/init.py)
```

3. 解决方案

需要根据报错提示，安装缺失的 te 依赖包

4.3 MindSpore 报错 Failed to combine the net and the parameters for param backbone.embedding.word_embedding.embedding_table.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 静态图

2. 报错信息

2.1 问题描述

context 设置 enable_alltoall=True 出现 AllToAllv 算子报错

2.2 报错信息

16 机 128 卡

```
- Framework Error Message: (For framework developers)------------------------------
----------------------Distribute Task Failed, error msg: davinci_model : load task
fail, return ret: 1343225860

------------------------------------------------------ C++ Call Stack: (For
framework developers)------------------------------------------------------
mindspore/ccsrc/plugin/device/ascend/hal/hardware/ascend_kernel_executor.cc:241
PreprocessBeforeRunGraphmindspore/ccsrc/plugin/device/ascend/hal/device/ascend_kern
el_runtime.cc:587
LoadTaskmindspore/ccsrc/plugin/device/ascend/hal/device/ge_runtime/task/hccl_task.c
c:111 Distribute

time stamp 2023.10.25-11:16:10 before loading ckpt[CRITICAL]
ME(1094:281473579920368,MainProcess):2023-10-25-11:16:10.522.225
[mindspore/train/serialization.py:116] Failed to combine the net and the parameters
for param backbone.embedding.word_embedding.embedding_table.
```

单机 8 卡：

```
- Framework Error Message: (For framework developers)------------------------------
----------------------Distribute Task Failed, error msg: davinci_model : load task
fail, return ret: 1343225860, the task is Default/AllToAllv-op16385_1532618

------------------------------------------------------ C++ Call Stack: (For
framework developers)------------------------------------------------------
mindspore/ccsrc/plugin/device/ascend/hal/hardware/ascend_kernel_executor.cc:198
PreprocessBeforeRunGraphmindspore/ccsrc/plugin/device/ascend/hal/device/ascend_kern
el_runtime.cc:587
LoadTaskmindspore/ccsrc/plugin/device/ascend/hal/device/ge_runtime/task/hccl_task.c
c:104 Distribute
```

3. 根因分析

报错出现 *Distribute Task Failed*，如果报错日志信息没有其他的 ERROR 或 CRITICAL 报错，需要 export GLOG_v=1 开启 info 日志，看看具体是哪个算子下发失败。

AlltoAllv 的算子报错：

1、查看脚本中 enable_alltoall 是否为 True，如果是，可改成 False 看一下是否还有这个报错；

2、如果改成 False，不报错，那就是通信方式修改导致的问题，可先改成 False 规避，但性能会很差；

3、dump 图，分析为什么会插 alltoall 算子（查看 hwopt_d_end_graph_*.ir 图）（mtp 平台保存图，需要创建输出（如 graphs）；并在任务创建输出引用（如名字为 output，关联创建的输出名 graphs）；修改脚本路径为/opt/huawei/quoteModel/output/***）

分析示例如下：

```
%1214 (eguiv[CNode]23017) A11ToA11v(%1212, %1213) primitive attrs:
(IsFeatureMapInputList: (0, 1),comm reuse: true, group:2-16453000547691086251,pri
format: NC1HWCO, recv_rank ids: (0, 1), send rar
       : (<Tensor[Float16], (1, 4096, 1, 8792)>, <Tensor[Float16], (1, 4096,
1,8792)>) -> (<Tuple[Tensor[Float16]*2], Tupleshape((1, 4096, 1, 8792), (1,4096, 1,
8792))>)LF
       : (<Float16xDefaultFormat[const vector][1, 4096, 1, 8792]>,
<Float16xDefaultFormat[const vector][1, 4096, 1, 8792]>) ->
(<Float16xDefaultFormat[const vector] [1, 4096, 1, 8792]>, <Float16xDefaultFor
       : (Tensor, Tensor) ->(TupleUnfold) LF
       # fullname_with_scope:(Default/A11ToA11v-op12859) LF
```

查看 alltoallv 的输入，往上追溯到非框架插的第一个算子，记住 alltoallv 的数据切分策略。

```
%1206 (equivlogits) - MatMul(%1198, %1205) (instance name:matmul} primitive_att rs:
{IsFeatureMapInputList: (0), input names: fx1, x21, pri format: FRACTAL NZ,
out_strategy: None, transpose x2: true, transpose b: true, visited: true, custaicpu:
mindspore cpukernels, tbe fusiontype: Matmul, is dynamic len: false, in strategy:
((1, 1). (8, 1)). output_names: [output], transpose a: false, IsFeatureMapOutput:
true,is kernel dynamie impl: false, transpose xl: false) cnode primal attrs:
(unique id: 905614}LF
       :  (<Tensor[Float16], (8192, 5120)>, <Tensor[Float16], (8792, 5120)>) ->
(<Tensor[Float16], (8192, 8792)>)LF
       :  (<Float16xFRACTAL NZ[const vector][320, 512, 16, 16]>,<Float16xFRACTAL
NZ[const vector][320, 550, 16, 16]>) -> (<Float16xFRACTAL NZ[const vector][550, 512,
16, 16]>)LF
       :  (Tensor, Tensor) ->(Tensor)LF
       # fullname with scope:
       (Default/network-
PanguAlphaTrainOneStepWithLossScaleCe11WithLossMask/network-
_VirtualDatasetCel1/_backbone-MicroBatchInterleaved/network-RrhfWithLoss/network-
PanguAlphaModel/head-PanGuHead/MatMul-op6643)LF
```

查看 alltoall 的输出，往下追溯到非框架插的第一个算子，记住 alltoallv 的数据切分策略。

```
$1228 (equivl0gprobs) 1 LogSoftmaxV2 (%1227) (instance name: loa softmaxl nrimit ve
attrs: (IsFeatureMapInputList: (0), pri_format: DefaultFormat,tbe_fusion_type:
Opaque, me op name: LogSoftmax, axis: -1, out strategy: None, is dynamıc ten: false,
```

```
in strategy: ((1, 1, 1)), visited: true, IsFeatureMapOutput: true, is
kerneldynamic_impl: true, op_adaptation_processed:
true)cnode_primal_attrs:{unique_id: 905407}LF
```

查看，两个算子的切分策略是否符合预期。一般能找到错误的切分策略，对应代码行，进行修改即可。

4. 解决方案

本次报错引入原因在于，alltoall 的输出算子切分策略为（1,1,1），不符合预期。对应代码，发现是切分没有指定策略，是默认按数据 dp 切分，修改即可

```
logprobs = self.log softmax(logits)/LF
logits = self.logprobs_from lables(logits[ : ,  : ,  : ] input_ids[ : , 1: ]) #:
```

4.4 MindSpore 训练报错 TypeError: Invalid Kernel Build Info! Kernel type: AICPU_KERNEL, node: Default/Concat-op1

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.0

执行模式： PyNative

2. 报错信息

2.1 报错信息

训练脚本报错：

```
TypeError: Invalid Kernel Build Info! Kernel type: AICPU_KERNEL, node:
Default/Concat-op1
```

开 pynative sync 之后，报错信息增加：

```
(<int64xdefaultformat>, <boolxdefaultformat>, <boolxdefaultformat>,
<boolxdefaultformat>, <boolxdefaultformat>, <boolxdefaultformat>,
<boolxdefaultformat>, <boolxdefaultformat>, object_type:
[Tensor,Tensor,Tensor,Tensor,Tensor,Tensor,Tensor,Tensor]) -> (<boolxdefaultformat>,
object_type: [Tensor], kernel_type: AICPU_KERNEL, op_type:
UNKNOWN_OP_TYPE)</boolxdefaultformat></boolxdefaultformat></boolxdefaultformat></bo
olxdefaultformat></boolxdefaultformat></boolxdefaultformat></boolxdefaultformat></b
oolxdefaultformat></int64xdefaultformat>
```

3. 根因分析

所有的 Concat 算子都不支持 Bool 类型的 Tensor。报错不清晰，根据日志内容无法获得类型不支持的根本错因。

4. 解决方案

将需要做 concat 的数据先转换为 int32，然后做 Concat 处理。例如：

```
list_int = []
   for item in list_int :
   list_int .append(mindspore.ops.Cast()(item, mindspore.int32))
   print(list_int)
```

4.5 MindSpore 模型正向报错 Sync stream failed:Ascend_0，plog 显示 Gather 算子越界

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.1

执行模式： 静态图

2. 报错信息

2.1 问题描述及报错

模型正向报错信息如下：

```
[ERROR] [2023-11-27 13:40:37] rank 49: Traceback (most recent call last):
  File "/home/ma-user/code/pangu nlp/src/obs.py", line 467, in wrapped func
    run func(cloud adapter) # main function
  File "/home/ma-user/code//pangu nlp/train.py", line 154, in run pangu\_train
    model.train(actual sink steps, train ds, callbacks-callbacks, sink
size=args.sink size, dataset sink mode=True)
  File "/home/ma-user/anaconda3/envs/python-3.9/lib/python3.9/site-
packages/mindspore/train/model.py", line 1068, in train
    self. train(epoch,
  File "/home/ma-user/anaconda3/envs/python-3.9/lib/python3.9/site-
packages/mindspore/train/model.py", line 114, in wrapper
    func(self, *args, **kwargs)
  File "/home/ma-user/anaconda3/envs/python-3.9/lib/python3.9/site-
packages/mindspore/train/model.py", line 623, in train
    self. train dataset sink process(epoch, train dataset, list callback,
  File "/home/ma-user/anaconda3/envs/python-3.9/1ib/python3.9/site-
packages/mindspore/train/model.py", line 708, in train
    dataset sink process outputs = train network(*inputs)
  File "/home/ma-user/anaconda3/envs/python-3.9/lib/python3.9/site-
packages/mindspore/nn/cell.py", line 680, in __call__
    out = self.compile and run(*args, **kwargs)
  File "/home/ma-user/anaconda3/envs/python-3.9/1ib/python3.9/site-
packages/mindspore/nn/cell.py", line 1023, in compile and run
    return cell graph\_executor(self, \*new args, phase-self.phase)
  File "/home/ma-user/anaconda3/envs/python-3.9/lib/python3.9/site-
packages/mindspore/common/api.py", line 1589, in __call__
    return self.run(obj, *args, phase=phase)
  File "/home/ma-user/anaconda3/envs/python-3.9/lib/python3.9/site-
packages/mindspore/common/api.py", line 1628, in run
    return self._exec_pip(obj, *args, phase=phase_real)
  File "/home/ma-user/anaconda3/envs/python-3.9/1ib/python3.9/site-
packages/mindspore/common/api.py", line 121, in wrapper
```

```
    results = fn(*arg, **kwargs)
  File "/home/ma-user/anaconda3/envs/python-3.9/lib/python3.9/site-
packages/mindspore/common/api.py", line 1608, in _exec_pip
    return self. graph executor(args, phase)
RuntimeError: Sync stream failed:Ascend_0
```

plog 报错信息：

```
[ERROR] RUNTIME(665015,python3):2023-11-27-14:49:47.448.557
[device_error_proc.cc:1138]666400 ProcessStarsCoreErrorInfo:The error from
device(chipId:0, dieId:0), serial number is 14, there is an aivec error exception,
core id is 46, error code = 0x800000, dump info: pc start: 0x1240c001b000, current:
0x1240c001b4f0, vec error info: 0xd1070f2790, mte error info: 0x6c030655c8, ifu
error info: 0x37fb88b75f040, ccu error info: 0xdb3b4c902c8d5dff, cube error info: 0,
biu error info: 0, aic error mask: 0x6500020bd000288, para base: 0x1240c00bf000.
[ERROR] RUNTIME(665015,python3):2023-11-27-14:49:47.448.584
[device_error_proc.cc:1150]666400 ProcessStarsCoreErrorInfo:report error
module_type=5, module_name=EZ9999
[ERROR] RUNTIME(665015,python3):2023-11-27-14:49:47.448.589
[device_error_proc.cc:1150]666400 ProcessStarsCoreErrorInfo:The extend info:
errcode:(0x800000, 0, 0) errorStr: The DDR address of the MTE instruction is out of
range. fixp_error0 info: 0x30655c8, fixp_error1 info: 0x6c fsmId:0, tslot:0,
thread:0, ctxid:0, blk:17, sublk:0, subErrType:4.
[ERROR] RUNTIME(665015,python3):2023-11-27-14:49:47.448.606 [stream.cc:3116]666400
EnterFailureAbort:stream_id=2 enter failure abort.
[ERROR] RUNTIME(665015,python3):2023-11-27-14:49:47.448.647 [stream.cc:1480]666400
GetError:Stream Synchronize failed, stream_id=2, retCode=0x91, [the model stream
execute failed].
[ERROR] RUNTIME(665015,python3):2023-11-27-14:49:47.448.656
[task_info.cc:4737]666400 ReportErrorInfoForModelExecuteTask:model execute error,
retCode=0x91, [the model stream execute failed].
[ERROR] RUNTIME(665015,python3):2023-11-27-14:49:47.448.674
[task_info.cc:4704]666400 PrintErrorInfoForModelExecuteTask:stream_id=2, task_id=3,
write_sq_id=5, read_sq_id=5, fsm_state=0, sqVirtualAddr=3884417024, head equal tail
flag=0.
[ERROR] RUNTIME(665015,python3):2023-11-27-14:49:47.448.682
[task_info.cc:4710]666400 PrintErrorInfoForModelExecuteTask:model execute task
failed, device_id=0, model stream_id=2, model task_id=3, flip_num=0, model_id=0,
first_task_id=0
[ERROR] RUNTIME(665015,python3):2023-11-27-14:49:47.448.775
[task_info.cc:1574]666400 PrintErrorInfoForDavinciTask:Aicore kernel execute failed,
device_id=0, stream_id=5, report_stream_id=2, task_id=0, flip_num=0, fault
kernel_name=00_1_Default/Gather-op177, program id=0, hash=12989916100239791524.
[ERROR] RUNTIME(665015,python3):2023-11-27-14:49:47.448.814
[task_info.cc:1516]666400 GetArgsInfo:[AIC_INFO] args(0 to 4) after
execute:0x124ee3e00000, 0x124f7ac00e00, 0x124f79e00200, 0x124180000000,
[ERROR] RUNTIME(665015,python3):2023-11-27-14:49:47.448.823
[task_info.cc:1519]666400 GetArgsInfo:print 1 Times totalLen=(4*8)Bytes,
argsSize=32
[ERROR] RUNTIME(665015,python3):2023-11-27-14:49:47.448.834
[task_info.cc:1578]666400 PrintErrorInfoForDavinciTask:[AIC_INFO] after
execute:args print end
[ERROR] RUNTIME(665015,python3):2023-11-27-14:49:47.448.860 [engine.cc:3646]666400
StarsResumeRtsq:stop scheduling in abort failure mode: stream_id=2, sq_id=2,
sq_head=4, task_id=4, taskType=14.
[ERROR] RUNTIME(665015,python3):2023-11-27-14:49:47.448.869 [engine.cc:3478]666400
```

```
SyncTask:stream is in failure abort and need to reclaim all, stream_id=2.
[ERROR] RUNTIME(665015,python3):2023-11-27-14:49:47.448.880 [stream.cc:1480]666400
GetError:Stream Synchronize failed, stream_id=2, retCode=0x91, [the model stream
execute failed].
[ERROR] RUNTIME(665015,python3):2023-11-27-14:49:47.448.885 [stream.cc:1483]666400
GetError:report error module_type=5, module_name=EZ9999
[ERROR] RUNTIME(665015,python3):2023-11-27-14:49:47.448.890 [stream.cc:1483]666400
GetError:Aicore kernel execute failed, device_id=0, stream_id=5, report_stream_id=2,
task_id=0, flip_num=0, fault kernel_name=00_1_Default/Gather-op177, program id=0,
hash=12989916100239791524.
[ERROR] RUNTIME(665015,python3):2023-11-27-14:49:47.448.906 [stream.cc:1483]666400
GetError:report error module_type=5, module_name=EZ9999
[ERROR] RUNTIME(665015,python3):2023-11-27-14:49:47.448.911 [stream.cc:1483]666400
GetError:[AIC_INFO] after execute:args print end
```

3. 根因分析

plog 显示 gather 算子执行失败，且有 The DDR address of the MTE instruction is out of range，即地址访问越界，这种情况一般是 gather 的索引越界了。nlp 领域里对输入做 embedding 时用到 gather 算子，怀疑是输入数据超过了 vocabsize。构造相同 shape 的全 1 输入数据，发现模型可以正常跑，说明是数据问题。检查数据发现 vocabsize 参数和数据集不匹配，比数据集的实际 vocabsize 小。

4. 解决方案

使用和数据集的数据范围匹配的 vocabsize。

4.6 model.train 报错 Exception in training: The input value must be int and must > 0, but got '0' with type 'int'.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.10

执行模式：Graph

2. 报错信息

```
Traceback (most recent cal1 last):
   File "/home/ma-user/modelarts/user-job-dir/code_v18/sigma_sft_train.py", line 15,
in <module>
       runrtrain_no_ pipeline(opt)
   File "/home/ma-user/modelarts/user-job-dir/code_v18/pangu_sigma/sft/sigmasft.py,
line 294, in run_train_no_pipeline
       model.build(train dataset=ds, sink size=args_opt.sinksize,
epoch=actual_epoch num)

ValueError: The input value must be int and must > 0, but got '0' with type 'int'.
rank:5, batch_size: 1 data_size:z, eod_reset:1
```

3. 根因分析

根据报错信息，找到报错的代码；发现报错信息集中在 mindrecord，这个时候需要检查一下数据量；看报错信息，显示 data_size:2，说明数据量只有两个，这个时候需要检查自己生成 ds 数据集的 shard_id 和 num_shard 参数设置的是多少。如果数据量确实很少，且是本机 8 卡时，可按如下配置：

```
dataset = ds.MindDataset(data[data_start_ index: ], columns_list=columns_list,
shuffle=shuffle, num_samples=num_samples, shard_id_rank // 8, num_shards=device_num
// 8)
```

根因：数据量太少，在创建 ds 数据集时，num_shards 配置不对导致。

4. 解决方案

解决方法：增大数据量。

4.7 数据集异常导致编译(model.build)或者训练(model.train)卡住

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.10

执行模式：PyNative/ Graph

2. 报错信息

训练任务卡在编译或者训练阶段，或者编译阶段执行时间很长

3. 根因分析

使用 mindspore.dataset.GeneratorDataset 接口传入的 source，如果采用 next 的迭代方式：

1. 在图编译阶段，如果开启了数据下沉，ms 首先要获取 dataset 的总迭代次数，如果__len__没实现，会完整遍历一次数据集获取总数，导致编译时间长或者是卡住。

2. ms 以捕获到 StopIteration 异常终止迭代，所以如果__next__方法没有抛出异常，会导致一直迭代，编译卡住。

```
class MyIterable:
    def __init__(self):
        self._index = 0
        self._data = np.random.sample((5, 2))
        self._label = np.random.sample((5, 1))

    def __next__(self):
        if self._index >= len(self._data):
            raise StopIteration
        else:
            item = (self._data[self._index], self._label[self._index])
            self._index += 1
            return item

    def __iter__ (self):
```

```
        self._index = 0
        eturn self

    def __len__(self):
        return len(self._data)
```

4. 解决方案

必须实现__len__方法和__next__方法，且迭代结束时__next__方法必须抛出StopIteration 异常通知 mindspore 迭代结束。

4.8 baichaun2-13b 在 Ascend910 上持续溢出

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend910

软件环境: MindSpore 版本: 2.2.10

执行模式： 不限

Python 版本: 3.8.15

操作系统平台: linux

2. 报错信息

1、baichaun2-13b 在 Ascend910 上持续溢出；

2、baichaun2-13b 加载 belle 数据集溢出 wiki 数据集不溢出。

3. 根因分析

1、根因是 baichaun2-13b 对 attention_mask 进行了 pad，attention mask 存在全 0 的无效行，pad_token 没有任何可注意对象，Ascend910 在不同核的机器上处理模板不一致，导致 flash_attention 出现了溢出。

```
class CausalMask(nn.Cell):
    r""" Get the Lower triangular matrix from the input_ids."""
    @_LogActiononce(m_logger=logger,
key='AttentionMask',no_warning=_get_parallel_mode() in (ParallelMode.STAND_ALONE,))
    def __init__(self, seq_length, compute -type=mstype.float16, is_dynamic=False,
pad_token_id=0, use_flash_attent ion=False):

    def construct(self, tokens=None, masks=None):
        """Forward process of the CausalMask"
            if tokens is not None:
                bs = self.shape(tokens)[O]
                seq_len = self.shape(tokens)[1]
                input_mask = self.cast(self. not_equal(tokens, self .pad_token_id),
self.dtype)
            else:
                bs = self.shape(masks)[o]
                seq_len = self.shape(masks)[1]
                input_mask = self.cast(masks,  self.dtype)
```

```
            shape_right = (bs, 1, seq_len)
            shape_left = (bs, seq_len, 1)
            # Mask the padded inputs
            mask_left = self.reshape(input_mask, shape_left)
            mask_right = self.re shape(input_mask, shape_right)
            attention_mask = self.bmm(mask_left, mask_right)
            if not self.is_dynamic:
                lower_traiangle = self. expand_dim(self . lower_triangle_mask,     0)
            else:
                lower_triangle_mask = self.slice( self.lower_triangle_mask, (0, 0),
(seq_len, seq_len), (1, 1))
                lower_traiangle = self.expand_dim(lower- triangle_mask, 0)
            # the returned shape is [bs, seq_length, seq_length]
            attention_mask = self.mul(at tention_mask, lower_traiangle)
            return attention_mask
```

2、belle 数据集处理生成过程因为文本比较短，对 sql_length 进行了 pad，wiki 数据集采用的拼接超过 sql_length 进行截断，不会 pad，在生成 attention_mask 上出现了全 0 行，非严格下三角。

4. 解决方案

将 attention_mask 改成严格下三角问题解决：

```
class CausalMask(nn.Cell):
    r""" Get the Lower triangular matrix from the input_ids."""
    @_LogActiononce(m_logger=logger,
key='AttentionMask',no_warning=_get_parallel_mode() in (ParallelMode.STAND_ALONE,))
    def __init__(self, seq_length, compute -type=mstype.float16, is_dynamic=False,
pad_token_id=0, use_flash_attent ion=False):

    def construct(self, tokens=None, masks=None):
        """Forward process of the CausalMask"
            if tokens is not None:
                bs = self.shape(tokens)[O]
                seq_len = self.shape(tokens)[1]
                input_mask = self.cast(self. not_equal(tokens, self .pad_token_id),
self.dtype)
            else:
                bs = self.shape(masks)[o]
                seq_len = self.shape(masks)[1]
                input_mask = self.cast(masks,  self.dtype)

            shape_right = (bs, 1, seq_len)
            # shape_left = (bs, seq_len, 1)
            # Mask the padded inputs
            mask_left = self.reshape(input_mask, shape_left)
            # mask_right = self.re shape(input_mask, shape_right)
            # attention_mask = self.bmm(mask_left, mask_right)
            if not self.is_dynamic:
                lower_traiangle = self. expand_dim(self . lower_triangle_mask,     0)
            else:
                lower_triangle_mask = self.slice( self.lower_triangle_mask, (0, 0),
(seq_len, seq_len), (1, 1))
                lower_traiangle = self.expand_dim(lower- triangle_mask, 0)
            # the returned shape is [bs, seq_length, seq_length]
```

```
        attention_mask = self.mul(at tention_mask, lower_traiangle)
        return attention_mask
```

4.9 MTP 数据集分布式读写锁死，Failed to execute the sql [SELECT NAME from SHARD NAME;] while verifying meta file, database is locked]

1. 报错信息

```
[ERROR] Failed to execute the sql[SELECT NAME from SHARD_NAME;] while verifying
meta file, database is locked.
Please check the meta file: /opt/huawei/schedule-train/output/data_set/sms_1w_
true.mindrecord.db.
```

2. 根因分析

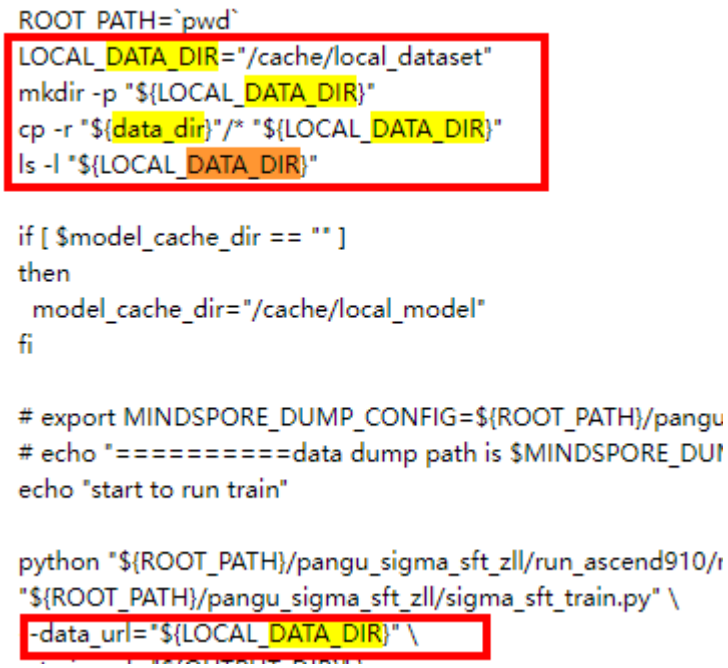
现在的数据路径，默认拉下来是 HDFS 盘，全局访问一个。规避办法是改变数据集路径，在拉任务的 shell 里面把数据集拷贝到/cache/下面，然后去访问这个路径。

4. 解决方案

1、创建数据集，将本地数据集，压缩成 zip 包，传上 mtp，成一个新的数据集，假设数据集名称叫 mindrecord_1020,内容为本地上传的 test.zip。

2、任务关联数据集，点击自己的任务名，进入，数据集进行添加。数据集目录名称命名为 data_dir，点击数据集名称的三个点，选择刚创建好的数据集：mindrecord_1020。

3、修改脚本：核心内容就是将数据集拷贝到/cache 下, cp -r "$(data_dir)"/*" "${LOCAL_DATA_DIR}"，这里的${data_dir}就是任务关联数据集的数据集目录名称，可自行修改。



4.10 MTP 任务卡死，平台报错信息['ROOT_CLUSTER'] job failed.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.10

执行模式： 不限

Python 版本: 3.8.15

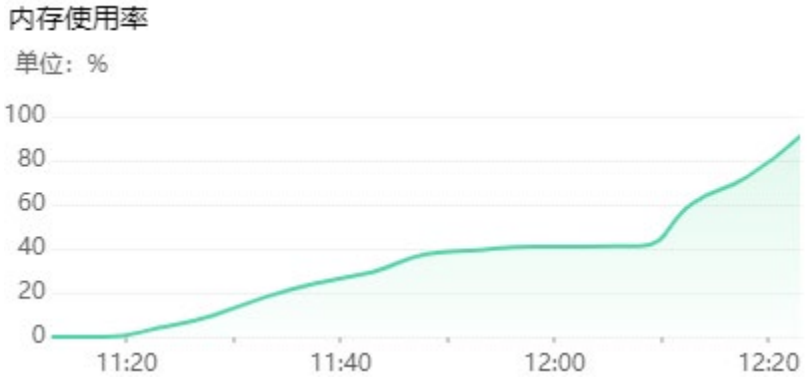
操作系统平台: linux

2. 报错信息

MTP 跑 sft 大模型过程中遇到任务卡死，无脚本报错，只有平台日志。

```
The parse job starts. Please wait.
['ROOT_CLUSTER', 'NODE_ANOMALY'] job failed. Please check the detail log.
The parse job is complete.
grep: /home/ma-user/modelarts/log/failure_analysis_result.json: No such file or
directory
```

3. 根因分析



联系平台确认,ModelArt 日志打印信息为 host 内存 oom 报错，报错信息如下：

```
['ROOT_CLUSTER'] job failed.
```

- 可以通过 psutil 工具打印 Host 内存利用率，监控内存信息：

```
import psutil

memory_info = psutil.virtual_memory()
print("总容量：{} GB".format(round(memory_info.total / 1024 / 1024 / 1024, 2)),
flush=True)
print("已用容量：{} GB".format(round(memory_info.used / 1024 / 1024 / 1024, 2)),
flush=True)
print("可用容量：{} GB".format(round(memory_info.free / 1024 / 1024 / 1024, 2)),
flush=True)
print("使用率：{} %".format(memory_info.percent), flush=True)
```

4. 解决方案

确认 host 内存溢出之后，通过打印日志判断是模型加载数据集过程中，缓存持续飙升导致 oom

1. 减小数据处理操作的缓冲队列大小（默认值：16）

   ```
   mindspore.dataset.config.set_prefetch_size(size)
   设置流水线中各个数据处理操作的缓冲队列大小。
   缓冲队列的存在使得当前操作在下一操作取走数据前就能开始处理后续数据，各操作异步并发地执行。
   更大的缓冲队列大小能够减少相邻操作吞吐速率不平衡时的整体处理时延，但也会消耗更大的系统内存。
   ```

2. 减少数据集处理过程中的 map 函数的使用也可以降低内存消耗。

4.11 大模型获取性能数据方法

MindSpore 提供环境变量使能或修改训练脚本两种方式来获取性能数据。

这里推荐直接修改训练脚本，该方法可灵活的适配数据下沉和非数据下沉等训练方式。

1、环境变量使能获取性能数据：

在运行网络脚本前，配置 Profiler 相关配置项。

```
export MS_PROFILER_OPTIONS='{"start": true, "output_path": "/XXX", "memory": false,
"hccl": false, "aicore_metrics": 0, "l2_cache": false}'
```

start (bool，必选) - 设置为”true”，表示使能 Profiler；设置成”false”，表示关闭性能数据收集，默认值：”false”。

output_path (str, 可选) - 表示输出数据的路径（绝对路径）。默认值：”./data”。

memory (bool，可选) - 表示是否收集 Tensor 内存数据。当值为”true”时，收集这些数据。默认值：”false”。

hccl (bool, 可选) - 表示是否在多设备训练中收集通信性能数据。当值为”true”时，收集这些数据。在单台设备训练中，该参数的设置无效。默认值：”false”。

aicore_metrics (int, 可选) - 设置 AI Core 指标类型，默认值：0。

l2_cache (bool, 可选) - 设置是否收集 l2 缓存数据，默认值：”false”。

2、修改网络训练脚本使能获取性能数据：

2.1 对于数据非下沉

如果用户是 for 循环格式进行模型训练，可参考以下代码获取性能数据

```
profiler = ms.Profiler(start_profile=False)
data_loader = ds.create_dict_iterator()

for i, data in enumerate(data_loader):
    train()
    if i==100:
        profiler.start()
    if i==200:
        profiler.stop()

profiler.analyse()
```

该代码主要需要包含以下功能：

1. 在模型训练之前初始化 profiler；

2. 在 for 循环中，自定义哪个 step 启动 profiler 工具；

3. 在 for 循环中，自定义哪个 step 关闭 profiler 工具；

4. 在 for 循环结束，调用 analyse 分析 profiler 数据。

完整代码参考：数据非下沉获取 profiling 数据完成代码样例

如果用户是 model.train 的方式进行模型训练，需要自定义获取性能数据的方法

用户可基于 step 开启，也可基于 epoch 开启，主要是需要自定义类，定义开启 profiler 和关闭 profiler 及 profiler 解析的函数。这里简单介绍介于 step 开启方法。

```
import mindspore as ms

class StopAtStep(ms.Callback):
    def __init__(self, start_step, stop_step):
        super(StopAtStep, self).__init__()
        self.start_step = start_step
        self.stop_step = stop_step
        self.profiler = ms.Profiler(start_profile=False)
```

```
    def step_begin(self, run_context):
        cb_params = run_context.original_args()
        step_num = cb_params.cur_step_num
        if step_num == self.start_step:
            self.profiler.start()

    def step_end(self, run_context):
        cb_params = run_context.original_args()
        step_num = cb_params.cur_step_num
        if step_num == self.stop_step:
            self.profiler.stop()

    def end(self, run_context):
        self.profiler.analyse()
```

2.2 对于数据下沉

只需要在模型训练之前初始化 profiler 工具；再在模型训练结束之后，调用 analyse 接口进行性能数据分析即可。

代码参考如下：

```
if __name__ == '__main__':
    ms.set_context(mode=ms.GRAPH_MODE, device_target="Ascend")

    # Init Profiler
    # Note that the Profiler should be initialized before model.train
    profiler = ms.Profiler(output_path='./profiler_data')

    # Train Model
    net = Net()
    train(net)

    # Profiler end
    profiler.analyse()
```

获取性能数据的内容，具体可参考官网 profiling。

4.12 模型解析性能数据

使用 profiler 工具获取性能数据之后，mindspore 提供 mindinsight 工具对性能数据进行解析。

在解析之前，用户需要在机器安装 mindinsight 工具，且完成属性配置，如果用户之前在机器上使用过 mindinsight 工具解析过性能数据，可直接跳转至“网页版可视化性能数据”。

安装 mindinsight 工具

> pip install mindinsight

完成 mindinsight 属性配置

> pip show mindinsight

> vim /root/anaconda3/envs/ms/lib/python3.7/site-packages/mindinsight/conf/constants.py（修改为本机 mindinsight 安装路径的 constants.py 文件）

```
#HOST = '127.0.0.1'
HOST = '8.92.9.109'  #修改了 host
# Allow to support cross originresource sharing(CORS) enable. Default is disate
# If enable coRs, supPoRT REQUEsT METHOos should enable 'opTtRaLmethod.
ENABLE_ CORS = True
SUPPORT REQUEST METHODS = {'POST','GET', 'PUT', 'DELETE', 'OPTIONS'}  #增加了 options
```

按上述方法修改 constants.py 即可（HOST 修改为本机 ip）。

启动 mindinsight

> mindinsight start --port 9006 --summary-base-dir /home/net/pro/

port 可修改为任意端口

summary-base-dir 修改为存放 profiler 数据的路径

网页版可视化性能数据

mindinsight start 之后，会给出一个网址，复制网址，网页打开即可

展示了性能数据总览页面，包含了迭代轨迹（Step Trace）、算子性能、数据准备性能和 Timeline 等组件的数据总体呈现。各组件展示的数据如下：

迭代轨迹：将训练 step 划分为几个阶段，统计每个阶段的耗时，按时间线进行展示；总览页展示了迭代轨迹图。

算子性能：统计单算子以及各算子类型的执行时间，进行排序展示；总览页中展示了各算子类型时间占比的饼状图。

数据准备性能：统计训练数据准备各阶段的性能情况；总览页中展示了各阶段性能可能存在瓶颈的 step 数目。

Timeline：按设备统计每个 stream 中 task 的耗时情况，在时间轴排列展示；总览页展示了 Timeline 中 stream 和 task 的汇总情况。可视化性能数据之后，用户可根据实际性能耗时场景跳转至对应的章节进行性能优化。

具体可参考官网 mindinsight。

4.13 数据集处理结果精细对比

数据集处理结果可视化

一般情况下，原始数据集会经过一系列数据增强操作，最终转换为网络的输入。如果数据增强操作有问题，可能会导致精度对不齐。

此时，我们可以将经过数据增强后的数据转换为可视化的数据，比如 cv 网络，可以转换成图片，并将 label 标记在图片上，观察增强后的数据是否合理，大致判断数据处理过程是否存在问题。

如果数据处理过程中不存在随机因素，可以使用如下步骤进行精细对比：

数据集处理结果精细对比

1、MindSpore 和对标网络都可使用 TroubleShooter 工具，将数据处理结果保存为 npy 文件。

示例代码：

```
import troubleshooter as ts
import mindspore as ms
from mindspore import nn, Tensor

class NetWorkSave(nn.Cell):
    def __init__(self, file):
        super(NetWorkSave, self).__init__()
        self.file = file

    def construct(self, x):
        ts.save(self.file, x)
        return x

x1 = Tensor(-0.5962, ms.float32)
net = NetWorkSave('/tmp/ms/ms_tensor')
net(x1)
# x1 被保存至/tmp/ms/0_ms_tensor.npy
```

2、使用 TroubleShooter 工具来对比两个 npy 文件的相似度。

示例代码：

```
import troubleshooter as ts
ts.migrator.compare_npy_dir("/tmp/ms", "/tmp/pt")
```

4.14 通过优化数据来加速训练速度

借助自动数据加速工具来优化性能

MindSpore 提供了一种自动数据调优的工具——Dataset AutoTune，用于在训练过程中根据环境资源的情况自动调整数据处理管道的并行度，最大化利用系统资源加速数据处理管道的处理速度。

启动自动数据加速方法、自动数据加速工具的约束及使用样例可参考：自动数据加速

1. 优化 batch 操作

在数据处理的最后阶段，会使用 batch 操作将多条数据组织成一个 batch，然后再传递给网络用于训练。

对于 batch 操作的性能优化可参考：数据 batch 性能优化

2. 优化数据增强

数据增强性能优化建议如下：

优先使用 MindSpore 提供的数据增强操作，能获得更好的性能，如果性能仍无法满足需求，可采取如下方式进行优化：

1、多线程优化：增大 map 接口的参数 num_parallel_workers（默认值：8）来取得更好的性能。

2、融合算子优化：在当前 CPU 占用率比较高时，使用融合操作来降低 CPU 占用会获得更好性能，可以通过配置环境变量 export OPTIMIZE=true 来使其生效。

3、Compose 优化：在当前 CPU 占用率比较高时（如：单机多卡训练），通过一个 map 操作接收多个增强操作（会按照顺序应用这些操作）来降低 CPU 降低竞争以取得更好性能

具体可参考：数据增强性能优化

3. 优化 shuffle 操作

shuffle 操作主要是对有序的数据集或者进行过 repeat 的数据集进行混洗。MindSpore 专门为用户提供了 shuffle 函数，它是基于内存缓存实现的，其中设定的 buffer_size 参数越大，混洗程度越大，但内存空间、时间消耗也会更大。

shuffle 性能优化建议如下：

1、直接使用数据集加载接口中的 shuffle=True 参数进行数据的混洗；

2、如果使用的是 shuffle 函数，当混洗效果无法满足需求，可通过调大 buffer_size 参数的值来优化混洗效果；当机器内存占用率过高时，可通过调小 buffer_size 参数的值来降低内存占用率。

具体可参考：数据 shuffle 性能优化

5 并行问题案例

5.1 MindSpore 分布式模型并行报错：operator Mul init failed 或者 CheckStrategy failed.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.1

执行模式： 静态图

2. 报错信息

2.1 问题描述

分布式模型并行，报错 operator Mul init failed 或者 CheckStrategy failed.

```
[ERROR] PARALLEL(99,ffff90ca1930,python):2023-07-10-13:29:46.150.073
[mindspore/ccsrc/frontend/parallel/ops_info/arithmetic_info.cc:89] CheckStrategy]
MulInfo7575 : Invalid strategy.
[ERROR] PARALLEL(99,ffff90ca1930,python):2023-07-10-13:29:46.150.127
[mindspore/ccsrc/frontend/parallel/ops_info/operator_info.cc:885]
InitForCostModelWithAutoRepeatCalc] MulInfo7575: CheckStrategy failed.
[ERROR] PARALLEL(99,ffff90ca1930,python):2023-07-10-13:29:46.150.140
[mindspore/ccsrc/frontend/parallel/ops_info/operator_info.cc:852] Init]
MulInfo7575 : Init failed.

----------------------------------------------------
- The Function Call Stack: (For framework developers)
----------------------------------------------------
In file /opt/huawei/schedule-train/algorithm/*/nn/transformer.py:1197/
current_key = self.mul1(self.tile(key, (1, 1, 1, self.src_seq_length)),/
In file /opt/huawei/schedule-train/algorithm/*/transformer.py:1187/          if
self.is_first_iteration:/
In file /opt/huawei/schedule-
train/algorithm/pangu_sigma/ms/parallel/nn*/angu_sigma/src/pangu_moe.py:133/
attention, layer_present = self.attention(query_vector, input_x, input_x,
abs_pos_embed, rel_pos_embed, input_mask,/
In file /opt/huawei/schedule-train/algorithm/*/main_model.py:379/
encoder_output, _ = self.top_query_layer(encoder_output, top_query_hidden_states,
encoder_masks,/
```

```
In file /opt/huawei/schedule-train/algorithm/*/main_model.py:370/       if
self.is_pipeline:/
In file /opt/huawei/schedule-train/algorithm/*/main_model.py:434/
output_states, word_table = self.backbone(input_ids, input_position, input_rel_pos,
attention_mask,/
In file /opt/huawei/schedule-train/algorithm/*/mian_model.py:530/       logits =
self.backbone(input_ids, position_ids, None, attention_mask, expert_ids,
current_index,/

----------------------------------------------------
- C++ Call Stack: (For framework developers)
----------------------------------------------------
mindspore/ccsrc/frontend/parallel/step_parallel.cc:1580 ExtractStrategyAndInit
```

3. 根因分析

模型并行中 Mul 算子的切分策略不对，导致该报错。我们开 Info 日志可以看到以下信息。

```
[INFO] PARALLEL(283,ffff95abcbf0,python):2023-07-10-17:11:31.981.235
[mindspore/ccsrc/fronted/parallel/step_parallel_utils.cc:1253] ExtractStrategt]
Extract information: strategy ((4, 8, 1, 1), (1, 1, 1,11))
```

我们结合报错信息中的调用栈可以知道报错的 Mul 算子的具体位置。然后调整下该算子的切分策略即可。

4. 解决方案

```
self.mul1 = P.Mul().shard(((1, 1, 1, 1),
                           (1, 1, 1, 1)))
```

遇到此类的 CheckStrategy failed 报错，尝试修改报错算子的切分策略，具体策略还需根据实际而定。

5.2 MindSpore 跑模型并行报错 ValueError: array split does not result in an equal division

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.1

执行模式： 静态图

2. 报错信息

2.1 问题描述

MindSpore 跑模型并行，随机初始化模型能够正常跑通，加载预训练模型报 numpy 的错误，但是根据调用栈可以看到是参数切分部分调过去。报错：

```
ValueError: array split does not result in an equal division
```

报错信息：

```
Traceback (most recent call last):
  File "/opt/huawei/schedule-train/algorithm/*/main.py", line 701, in <module>
    main(config_)
  File "/opt/huawei/schedule-train/algorithm/*/main.py", line 673, in main
    train_prompt, train_actor_logprobs, train_sft_logprobs, train_critic_r,
train_reward_r)
  File "/home/ma-user/anaconda3/envs/MindSpore/lib/python3.7/site-
packages/mindspore/nn/cell.py", line 636, in __call__
    out = self.compile_and_run(*args, **kwargs)
  File "/home/ma-user/anaconda3/envs/MindSpore/lib/python3.7/site-
packages/mindspore/nn/cell.py", line 959, in compile_and_run
    self.compile(*args, **kwargs)
  File "/home/ma-user/anaconda3/envs/MindSpore/lib/python3.7/site-
packages/mindspore/nn/cell.py", line 936, in compile
    jit_config_dict=self._jit_config_dict, *args, **kwargs)
  File "/home/ma-user/anaconda3/envs/MindSpore/lib/python3.7/site-
packages/mindspore/common/api.py", line 1385, in compile
    result = self._graph_executor.compile(obj, args, kwargs, phase,
self._use_vm_mode())
  File "/home/ma-user/anaconda3/envs/MindSpore/lib/python3.7/site-
packages/mindspore/parallel/_utils.py", line 105, in _slice_parameter
    new_tensor = _load_tensor_by_layout(parameter, layout)
  File "/home/ma-user/anaconda3/envs/MindSpore/lib/python3.7/site-
packages/mindspore/parallel/_tensor.py", line 261, in _load_tensor_by_layout
    tensor_slice = np.split(tensor_slice, size)[rank]
  File "<__array_function__ internals>", line 6, in split
  File "/home/ma-user/anaconda3/envs/MindSpore/lib/python3.7/site-
packages/numpy/lib/shape_base.py", line 873, in split
    'array split does not result in an equal division') from None
ValueError: array split does not result in an equal division
```

3. 根因分析

在分析之前，首先需要知道模型并行加载切分的模型文件是需要首先编译才能够加载，以下是我们报错场景的一段伪代码：

```
net_with_grade = TrainOneStepWithLossScale(with_loss,**args)
token_idx = Tensor(np.random.random((8,4096)).astype(np.int32))
atten_mask = Tensor(np.random.sample((8, 4096, 4096)).astype(np.float32))
net_with_grade.compile(token_idx, atten_mask)

ckpt_name = f"node_{int(rank/8)}/saved_{rank}_large.ckpt"
param_dict = mindspore.load_checkpoint(local_ckpt_path)
train_not_load = mindspore.load_param_into_net(with_loss, param_dict)

dataloader = create_data(**args)
for data in dataloader.create_dict_iterator():
    train_token_idx = data['token_idx']
    train_atten_mask = data['atten_mask']
    loss = net_with_grade(train_token_idx, train_atten_mask)
    print(loss)
```

正如前述，加载预训练模型之后报错，我们在报错位置打印出了当时的 shape 信息，如下

```
_load_tensor_by_layout(parameter, layout): Parameter
(name=backbone.backbone.blocks.0.attention.projection.weight, shape=(80, 5120),
dtype=Float16, requires_grad=True) ([8, 8], [0, -1], [80, 5120], 0, True, '8-
682084589588865485')
(10, 5120) 8ba
```

根据报错是 slpit 的输入的 shape 不能被 size 整除，所以报错，我们 Parameter 原始 shape 信息是(640,5120), mp(模型并行数)是 8。所以我们编译之后会得到上面的(80,5120)的 parameter shape。但是报错时成了（10， 5120）显然是经过了二次切分导致。预期只需要一次切分即可。也就是说上面代码中 compile 会执行一次编译，在 for 循环时 net_with_grade(train_token_idx, train_atten_mask)又会执行一次编译导致出现二次切分。

4. 解决方案

对于解决此类问题，首先需要解决二次编译的问题，一般情况只要输入 shape 和数据类型一致就不会导致二次编图，这里出现重新编图可以排查数据输入 shape 和类型是否不一致。经过排查是 compile 时的 atten_mask 和训练时的 train_atten_mask 数据类型不一致导致。compile 时是 fp32,训练时是 fp16,修改 compile 时的第二个输入类型为 fp16,修改如下：

```
atten_mask = Tensor(np.random.sample((8, 4096, 4096)).astype(np.float16))
```

5.3 MindSpore 训练大模型报错：BrokenPipeError: [Errno 32] Broken pipe, EOFError

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.1.1

执行模式： 静态图

2. 报错信息

2.1 问题描述

使用 MindSpore 跑大模型报以下错误：

```
Exception in thread Thread-1:
Traceback (most recent call last):
  File "/usr/local/python3.9/lib/python3.9/threading.py", line 973, in
_bootstrap_inner
Exception in thread Thread-2:
Traceback (most recent call last):
  File "/usr/local/python3.9/lib/python3.9/threading.py", line 973, in
_bootstrap_inner
    self.run()
  File "/usr/local/Ascend/ascend-toolkit/latest/python/site-
packages/tbe/common/repository_manager/utils/multiprocess_util.py", line 91, in run
    self.run()
  File "/usr/local/python3.9/lib/python3.9/threading.py", line 910, in run
    key, func, args, kwargs = self.task_q.get(timeout=TIMEOUT)
  File "<string>", line 2, in get
```

```
    self._target(*self._args, **self._kwargs)
  File "/usr/local/python3.9/lib/python3.9/multiprocessing/pool.py", line 513, in
_handle_workers
    cls._maintain_pool(ctx, Process, processes, pool, inqueue,
  File "/usr/local/python3.9/lib/python3.9/multiprocessing/pool.py", line 337, in
_maintain_pool
    Pool._repopulate_pool_static(ctx, Process, processes, pool,
  File "/usr/local/python3.9/lib/python3.9/multiprocessing/pool.py", line 326, in
_repopulate_pool_static
  File "/usr/local/python3.9/lib/python3.9/multiprocessing/managers.py", line 809,
in _callmethod
    w.start()
  File "/usr/local/python3.9/lib/python3.9/multiprocessing/process.py", line 121,
in start
    self._popen = self._Popen(self)
  File "/usr/local/python3.9/lib/python3.9/multiprocessing/context.py", line 291,
in _Popen
    conn.send((self._id, methodname, args, kwds))
  File "/usr/local/python3.9/lib/python3.9/multiprocessing/connection.py", line 211,
in send
    return Popen(process_obj)
  File "/usr/local/python3.9/lib/python3.9/multiprocessing/popen_forkserver.py",
line 35, in __init__
    self._send_bytes(_ForkingPickler.dumps(obj))
  File "/usr/local/python3.9/lib/python3.9/multiprocessing/connection.py", line 416,
in _send_bytes
    super().__init__(process_obj)
  File "/usr/local/python3.9/lib/python3.9/multiprocessing/popen_fork.py", line 19,
in __init__
    self._send(header + buf)
     File "/usr/local/python3.9/lib/python3.9/multiprocessing/connection.py", line
373, in _send
self._launch(process_obj)
     File "/usr/local/python3.9/lib/python3.9/multiprocessing/popen_forkserver.py",
line 58, in _launch
n = write(self._handle, buf)
BrokenPipeError: [Errno 32] Broken pipe
    f.write(buf.getbuffer())
BrokenPipeError: [Errno 32] Broken pipe
Exception in thread Thread-1:
Traceback (most recent call last):
  File "/usr/local/python3.9/lib/python3.9/threading.py", line 973, in
_bootstrap_inner
    self.run()
  File "/usr/local/Ascend/ascend-toolkit/latest/python/site-
packages/tbe/common/repository_manager/utils/multiprocess_util.py", line 91, in run
Exception in thread Thread-1:
Traceback (most recent call last):
  File "/usr/local/python3.9/lib/python3.9/threading.py", line 973, in
_bootstrap_inner
Exception in thread Thread-1:
Traceback (most recent call last):
  File "/usr/local/python3.9/lib/python3.9/threading.py", line 973, in
_bootstrap_inner
    self.run()
```

```
  File "/usr/local/Ascend/ascend-toolkit/latest/python/site-
packages/tbe/common/repository_manager/utils/multiprocess_util.py", line 91, in run
    key, func, args, kwargs = self.task_q.get(timeout=TIMEOUT)
  File "<string>", line 2, in get
  File "/usr/local/python3.9/lib/python3.9/multiprocessing/managers.py", line 810,
in _callmethod
    kind, result = conn.recv()
  File "/usr/local/python3.9/lib/python3.9/multiprocessing/connection.py", line 255,
in recv
    buf = self._recv_bytes()
  File "/usr/local/python3.9/lib/python3.9/multiprocessing/connection.py", line 419,
in _recv_bytes
    buf = self._recv(4)
  File "/usr/local/python3.9/lib/python3.9/multiprocessing/connection.py", line 388,
in _recv
    raise EOFError
EOFError
```

3. 根因分析

预训练模型太大，导致在加载模型的时候 host 内存消耗完毕，系统会选择性清理一些进程，以释放一些被占用的内存，导致报此错误。

4. 解决方案

建议排查方向：host 内存是否占满，以及内存耗光的原因。

5.4 MindSpore 大模型并行需要在对应的 yaml 里面做哪些配置

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 报错信息

2.1 问题描述

MindSpore 大模型并行需要在对应的 yaml 里面做哪些配置

3. 解决方案

1. auto_trans_ckpt: True；

2. load_checkpoint: "" 路径到文件夹，模型并行需要把模型放在 rank_0 下面；

3. 需要把 mindformer/core/parallel_config.py 下面的 vocab_emb_dp 那一行注释掉；

4. 使用 pipeline 并行的时候，要求 micro_batch_num>=pipeline_stage；

5. 模型并行 mp 一般设置小一点，建议为 2，如果设置过大可能存在通信问题。

```
#load_checkpoint: "/home/wizardcoder/1_wizardcoder-mindformers/outpu t/checkpoint/"
# 权重需要放在这个文件的 rank_0 下面：：

auto_trans_ckpt: True  # If true, auto transform load_checkpoint to load in
distributed model

parallel_config:
  data_parallel: 1  # 4
  model_parallel: 1  # 8
  pipeline_stage: 8
  optimizer_shard: True
  micro_batch_num: 8
  vocab_emb_dp: True
  gradient_aggregation_group: 4
```

5.5 模型并行显示内存溢出

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 报错信息

2.1 问题描述

模型并行显示内存溢出

3. 根因分析

一般是因为模型太大。

4. 解决方案

需要用多卡或者多机跑，整体 HBM 至少模型大小的 4 倍。

5.6 MindSpore 模型 Pipeline 并行发现有些卡的 log 中 loss 为 0

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 报错信息

2.1 问题描述

MindSpore 模型 Pipeline 并行发现有些卡的 log 中 loss 为 0

3. 根因分析

该情况下可能不是问题，pipeline 并行需要查看最后一张卡的 loss，其余卡的 loss 均为 0。

5.7 MindSpore8 卡报 Socket times out 问题

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 报错信息

2.1 问题描述

MindSpore 跑 wizardcoder 模型，dp:mp:pp=1:2:4，num_layer=16 时可以跑通，但是当设置 num_layer=20 时出现如下报错。

2.2 报错信息

```
EI0006: Getting s0cket times out. Reason: 1. The remote does not initiate a connect
request. some NPUs in the cluster are abnormmal. 2. The remote does not initiate a
connect request because the collective communication operator is started too late
or is not sta rted by some NPU in the cluster.  3. The communication link is
disconnected. (For example, the IP addresses are not on the same network segment or
theTLS configurations are inconsistent.)
    Solution: 1. Check the rank service processes with othererrors or no errors in
the cluster.2. If this error is reported for all NPUs, check whether the time
difference be tween the earliest and latest errors is greater than the connect
timeout interval (120s by default). If so, adjust the timeout interval by using the
HCCL CONNECT TIMEOUT environment variable.3. Check the connectivity of the
communication link between nodes. (For details, see the TLS command andHCCN
connectivity check examples.). For details:https://www.hiascend.com/document
```

3. 解决方案

因此在多机并行时，只能采用离线切分的方法。具体操作如下：

1. 首先找出哪张卡没有报错 EI0006: getting socket time out，没有这个报错的卡对应的报错才是根本原因。例如本案例发现 rank1 没有报错，rank4 有报错。

rank1 不报错

```
[root@localhost research]# grep -rna "EI0006" /home/wizardcoder/1_wizardcoder-
mindformers/research/output/log/ rank_1/mindformer.log
6266: [WARNING] GE (1253271,python) :2023-09-08-16:36:48.256.311
[ error_manager.cc:510]1253271 ParseJsonFile: [Check][Config]There are the
sameerror code EI0006 in /us r/local/c30_0818/CANN-7.0/aarch64 -
linux/lib64/../conf/error_manager/error_code.json
```

rank4 报错

```
[root@localhost research]# grep -rna "EI0006" /home/wizardcoder/1_wizardcoder-
mindformers/research/output/log/rank_4/mindformer.log
EI0006: Getting s0cket times out. Reason: 1. The remote does not initiate a connect
request. some NPUs in the cluster are abnormmal. 2. The remote does not initiate a
connect request because the collective communication operator is started too late
or is not sta rted by some NPU in the cluster.  3. The communication link is
disconnected. (For example, the IP addresses are not on the same network segment or
theTLS configurations are inconsistent.)
    Solution: 1. Check the rank service processes with othererrors or no errors in
the cluster.2. If this error is reported for all NPUs, check whether the time
difference be tween the earliest and latest errors is greater than the connect
timeout interval (120s by default). If so, adjust the timeout interval by using the
HCCL CONNECT TIMEOUT environment variable.3. Check the connectivity of the
communication link between nodes. (For details, see the TLS command andHCCN
connectivity check examples.). For details:https://www.hiascend.com/document
```

2. 查看 rank1 的 mind.log 日志，发现是内存不足导致的问题，至此找到了问题原因。

```
EL0004: Failed to allocate memory.
    Possible Cause: Available memory is insufficient
    Solution: Close applications not in use.
    TraceBack (most recent call last):
    rtMalloc execute failed, reason=[driver error:out of memory]
[FUNC:FuncErrorReason][FILE:error_message_manage.cc][LINE:50]
    Call rtMalloc fail, purpose:feature map „used for op input and output,
size:54966232576, device_id:0[FUNC:MallocMemoryl[FILE: graph_mem_allocator.cc]
[LINE:74]
    Malloc Memory fail, purpose: feature map,used for op input and output,
memory_key:0_f, memory_size:54966232576, device_id:0[FǓNC:MallocMemoryl
[FILE:graph_mem_alloca  tor.cc][LINE: 128]
    Malloc featuremap memory fail. malloced_memory_sizel0] < mem_size [54966232576],
device_id[0] [FUNC:MallocFeatureMapMem] [FILE:davinci_model . cc][LINE:5294]
    MallocFeatureMapMem fail , data_size:54966232576, mode l_id:3, check invalidl
FUNC: InitFeatureMapAndP2PMem] [FILE: davinci_modeL.cc] [LINE:383]
```

5.8 MindSpore 模型 Pipeline 并行训练报错 RuntimeError: Stage 0 should has at least 1 parameter. but got none.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.1

执行模式： 不限

2. 报错信息

2.1 问题描述

在新机器上重装 Ascend+MindSpore2.1 环境，并行设置 dp:mp:pp = 1:1:2 时，报如下错误：（只要是设置了 pp>1，都报下面的错误）

2.2 报错信息

```
Traceback (most recent call last):
  File "wizardcoder/run_wizardcoder.py", line 149, in <module>
    device_id=args.device_id)
  File "wizardcoder/run_wizardcoder.py", line 81, in main
    task.train(train_checkpo int=ckpt, resume=resume)
  File "/home/wizardcoder/1_wizardcoder-mindformers/mind
formers/trainer/trainer.py", line 424, in train
    is_full _config=True, **kwargs)
  File " /home/wizardcoder/1_wizardcoder- mindformers/mindformers/trainer/caus
al_language_modeling/causal_l anguage_modeling.py", line 104, in train
    **kwargs)
  File "/home/wizardcoder/1_wizardcoder -mindformers/mindformers/trainer/base_t
rainer.py", line 631, in training_process
    initial_epoch=config.runner_config.initial_epoch)
  File "/root/miniconda3/envs /zxw/lib/python3.7/site-packages
/mindspore/train/model .py", line 1066, in train
    initial_epoch=initial_epoch)
  File "/root/miniconda3/envs/zxw/lib/python3.7/site-pack
ages/mindspore/train/model.py", line 113, in wrapper
    func(self, *args, **kwargs)
  File "/root/miniconda3/envs/zxw/lib/python3.7/site-
packages/mindspore/train/model.py", line 620, in _train
    cb_params, sink_size,initial epoch, valid infos)
  File "/root/miniconda3/envs/zxw/lib/python3.7/site-pack
ages/mindspore/train/model .py", line 703, in _train_dataset_s ink_process
    outputs = train_network(*inputs)
  File "/root/miniconda3/envs/zxw/lib/python3.7/site-pack
ages/mindspore/nn/cell.py", line 637, in _call_
    out = self.compile and run(*args **kwargs)
  File "/root/miniconda3/envs/zxw/lib/python3.7/site-packages
/mindspore/nn/cell.py", line 961, in compile_and_run
    self.compile(*args, **kwargs)
    File "/root/miniconda3/envs/zxw/l ib/python3.7/site-packages/mindspo
re/nn/cell.py", line 939, in compile
    jit_config dict=self.j it config_ dict, *compile_args, **kwargs)
  File "/rootyminiconda3/envs7z xw/lib/python3.7/site-packages/mindspo
re/common/api.py", line 1623, in compile
    result = self. _graph executor.compile(obj, args, kwargs, phase,
self._use_vm_mode())
```

3. 根因分析

应该是开启了 cell 共享的环境变量，所以报这个 pipeline 错误。

4. 解决方案

两种解决方法：

1. 关闭 cell 共享环境变量，使用命令 export MS_DEV_CELL_REUSE=0。Cell 共享的作用是可优化编译性能。

2. 如果不关闭 cell 共享，则需要配置相应的装饰器。

```
from mindformers.models.utils import cell_reuse
from mindformers.modules. transformer.moe import default_moe_config
```

```
from mindformers.modules.layers import LayerNorm, Dropout
from mindformers.core.loss import CrossEntropyLoss

@MindFormerRegister.register(MindFormerModuleType. MODELS)
class WizardCoderLMHeadModel (BaseModel):
    r"""..."""
    ...
    @cell_reuse()
    def _init_(self, config: WizardCoderConfig = None):
        config = config if configis not None else WizardCoderConfig()
        super(WizardCoderLMHeadModel,self).__init__(config, auto_prefix=True)
```

【注意】：MindSpore2.2.0 版本，cell_reuse 装饰器的写法有变化，不需要后面的括号。

```
@MindFormerRegister.register(MindFormerModuleType. MODELS)
class WizardCoderLMHeadModel (BaseModel):
    r"""..."""
    ...
    @cell_reuse
    def _init_(self, config: WizardCoderConfig = None):
        config = config if configis not None else WizardCoderConfig()
        super(WizardCoderLMHeadModel,self).__init__(config, auto_prefix=True)
```

5.9 并行策略为 8:1:1 时报错 RuntimeError: May you need to check if the batch size etc. in your 'net' and 'parameter dict' are same.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 报错信息

2.1 问题描述

只开数据并行的时候，出现报错。

2.2 报错信息

```
RuntimeError: For 'load_pa ram_into_net', backbone.blocks .0.attention.project
ion.weight in the argument 'net' should have thesame shape as
backbone.blocks .0.attention.projection.weight in the argument 'parameter dict' But
got its shape (768, 6144) in the argument 'net' and shape (6144, 6144) in the
argument 'parameter_dict'.May you need to check whether the checkpoint you loadod
is correct or the batch size and so on in the 'net' and 'parameter dict' are same.
```

3. 根因分析

分析可知，在数据并行的时候也自动开启了模型切分，这是因为配置参数 optimizer_shard 设为了 True，将其设为 False 后。即可关闭数据并行模式下的模型自动切分。

4. 解决方案

如果想要使用 optimizer_shard，则在数据并行模式下也需要先进行模型切分。

重点：在设置上述参数后发现，loss_scale 出现很大问题（正常情况溢出 loss_scale 最低到 1 就停止降低，但是当前出现异常 loss_scale 更新到 unavailable。

只有数据并行时，将 parallel_mode 设为 0（即数据并行模式），此时 optimizer_shard 设为 True 也不影响使用，full_batch 参数必须设为 False，gradients_mean 建议设为 True。

5.10 模型并行策略为 1:1:8 时报错 RuntimeError: Stage num is 8 is not equal to stage used: 5

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.0

执行模式： 不限

2. 报错信息

2.1 问题描述

模型跑 4 层网络，设置并行策略为 dp:mp:pp=1:1:8，出现报错

2.2 报错信息

```
Traceback (most recent call last):
  File "wizardcoder/run_wizardcode r.py", line 148, in <module>
    device_id=args.device_id)
  File "wizardcoder/run_wizardcoder.py", line 90, in main
    task. finetune(finetune_checkpoint=config.load_checkpoint
auto_trans_ckpt=config .auto_trans_ckpt, resume=resume)
  File "/home/wizardcoder/1_wizardcoder-mindformers-
916/mindformers/trainer/trainer.py", Tine 522, in finetune
    is_full_config=True, **kwargs
  File"/home/wizardcoder/1_wizardcoder-mindformers-916/mindformers/traine r/caus
al_language_modeling/caus al_language_modeling.py", line 106, in train
    **kwargs)
  File "/home/wizardcoder/1_wizardcoder-mindformers-916/mindformers/traine
r/base_trainer.py", line 616, in training_process
    transform_and_load_checkpoint (config, model, network, dataset)
  File "/home/wizardcoder/1_wizardc oder-mindformers-916/mindforme
rs/trainer/utils.py", line 300, in transform_ and_load_checkpoint
    build_model(config, model, dataset, do_eval=do_eval, do_predict=do_predict)
  File "/home/wizardcoder/1_wizardcoder-mindformers-
916/mindformers/trainer/utils.py", line 330, in build_model
    sink_size=config.runner_config.sink_size)
```

```
  File "/root/miniconda3/envs/wiz ardcoder/lib/python3.7/site-
packages/mindspore/train/model .py", line 1263, in build
    self._init(train_dataset , valid_dataset, sink_siz e, epoch)
  File "/root/miniconda3/envs/wiza rdcoder/lib/python3.1/site-packages
/mindspore/train/model.py", line 524, in _init
    train_network.compile( *inputs)
  File "/root/miniconda3/envs/wizardcoder/lib/python3 .7/site-
packages/mindspore/nn/cell .py", line 939, in compile
    jit_config_dict=self._jit_config_dict, *compile_args, **kwargs)
  File n/root/miniconda3/envs7wiza rdcoder/lib/python3.7/site-packages
/mindspore/common/api.py", line 1623, in compile
    result = self. araoh executor.comoilelobi. aras. kwaras, phase, self. use vm
mode()
RuntimeError: Stage num is 8 is not equal to stage used: 5
```

3. 根因分析

这是因为模型层数只有 4 层，无法进行 pipeline=8 的分层切割。

4. 解决方案

需要满足 pipeline_stage 小于等于 num_layers 这一条件。

5.11 MindSpore 报错：wq.weight in the argument 'net' should have the same shape as wq.weight in the argument 'parameter_dict'.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.3

执行模式： Graph

2. 报错信息

执行分布式训练任务，加载 ckpt 出现如下报错。

```
For 'load_param_into_net', wq.weight in the argument 'net' should have the same
shape as wq.weight  in the argument 'parameter_dict'. But got its shape (512, 8192)
in the argument 'net' and shape (8192, 8192) in the argument 'parameter_dict'. May
you need to check whether the checkpoint you loaded is correct or the batch size
and so on in the 'net' and 'parameter_dict' are same.
```

3. 根因分析

a. 错误出现的位置为，加载 checkpoint 时失败。

b. 报错信息为：网络中 wq.weight 参数的 shape 为(512, 8192)，但是 checkpoint 文件中 q.weight 参数的 shape 为(8192, 8192)，shape 不一致加载失败了。

c. 排查是网络的 shape 还是 checkpoint 文件的 shape 不符合预期，经过排查发现，网络中的 shape 是切分后的，是符合预期的。

d. 发现 checkpoint 中保存的参数 shape 是全量的，并不是切分好的。

e. 进一步排查发现，save_checkpoint 方法默认会把所有 parameter 合并保存成全量的，导致和网络中 shape 不一致。

4. 解决方案

在调用 save_checkpoint 的时候，把 integrated_save 参数手动设置为 false，使所有节点保存切分后的参数。重新加载保存后的 checkpoint 文件，问题解决 。

5.12 MindSpore 跑分布式报错 TypeError: The parameters number of the function is 636, but the number of provided arguments is 635.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.0&2.2.10

执行模式： Graph

2. 报错信息

在跑分布式推理时，调用了 infer_predict_layout 报参数数目对不上。

```
TypeError: The parameters number of the function is 636, but the number of provided
arguments is 635.
```

脚本代码

```
#调用方式
# shard model and load sharded ckpt
warm_up_model = Model(model)
warm_up_model.infer_predict_layout(ms.Tensor(np.ones(shape=(1, 3, 336,336)),
ms.float32))
# 定义
class Blip2ImageToTextGeneration(Blip2Llama):
    """
    Blip2ImageToTextGeneration rely on Blip2Llama, used for image to text
genearation.
    Args:
        config (Blip2Config): The config of Blip2ImageToTextGeneration.
    Examples:
        >>> from mindformers import Blip2ImageToTextGeneration
        >>> model =
Blip2ImageToTextGeneration.from_pretrained('itt_blip2_stage2_vit_g_llama_7b')
        >>> type(model)
        <class 'mindformers.models.blip2.blip2_llama.Blip2ImageToTextGeneration'>
    """
    _support_list = MindFormerBook.get_model_support_list()['itt']['blip2']
    def __init__(self, config: Blip2Config, **kwargs):
        super(Blip2ImageToTextGeneration, self).__init__(config, **kwargs)
        self.llama_model.set_train(False)
```

```
        self.one_prefix = ops.Ones()
        self.expand_dims = P.ExpandDims()
        self.query_length = self.config.qformer_config.query_length
    def construct(self, image: ms.Tensor, text_input_ids: ms.Tensor):
        if len(text_input_ids.shape) == 1:
            text_input_ids = self.expand_dims(text_input_ids, 0)
       ............
```

报错信息

```
Traceback (most recent call last):
  File "multi_gpu_infer.py", line 346, in <module>
    eval_model(args)
  File "multi_gpu_infer.py", line 260, in eval_model
    warm_up_model.infer_predict_layout(ms.Tensor(np.ones(shape=(1, 3, 336,336)),
ms.float16))
  File "/root/miniconda3/envs/MindSpore/lib/python3.7/site-
packages/mindspore/train/model.py", line 1882, in infer_predict_layout
    predict_net.compile(*predict_data)
  File "/root/miniconda3/envs/MindSpore/lib/python3.7/site-
packages/mindspore/nn/cell.py", line 998, in compile
    jit_config_dict=self._jit_config_dict, *args, **kwargs)
  File "/root/miniconda3/envs/MindSpore/lib/python3.7/site-
packages/mindspore/common/api.py", line 1547, in compile
    result = self._graph_executor.compile(obj, args, kwargs, phase,
self._use_vm_mode())
TypeError: The parameters number of the function is 636, but the number of provided
arguments is 635.
FunctionGraph :
mindformers_models_blip2_blip2_llama_Blip2ImageToTextGeneration_construct_1
NodeInfo: In file
/disk1/fail/scripts/mf_parallel0/mindformers/models/blip2/blip2_llama.py:536
    def construct(self, image: ms.Tensor, text_input_ids: ms.Tensor):
    ^
----------------------------------------------------
- C++ Call Stack: (For framework developers)
----------------------------------------------------
mindspore/ccsrc/pipeline/jit/ps/static_analysis/evaluator.cc:486 Eval
----------------------------------------------------
- The Traceback of Net Construct Code:
----------------------------------------------------
# In file
/disk1/fail/scripts/mf_parallel0/mindformers/models/blip2/blip2_llama.py:536
    def construct(self, image: ms.Tensor, text_input_ids: ms.Tensor):
    ^
# In file
/disk1/fail/scripts/mf_parallel0/mindformers/models/blip2/blip2_llama.py:544
        projected_qformer_output = self.forward_qformer_and_proj(image)
                                   ^
# In file
/disk1/fail/scripts/mf_parallel0/mindformers/models/blip2/blip2_llama.py:484
    def forward_qformer_and_proj(self, image: ms.Tensor):
    ^
# In file
/disk1/fail/scripts/mf_parallel0/mindformers/models/blip2/blip2_llama.py:486
        image_embeds = self.visual_encoder(image)
```

```
                    ^
# In file /disk1/fail/scripts/mf_parallel0/mindformers/models/blip2/blip2.py:112
    def construct(self, image):
# In file /disk1/fail/scripts/mf_parallel0/mindformers/models/blip2/blip2.py:113
        return self.construct_without_pool(image)
               ^
# In file /disk1/fail/scripts/mf_parallel0/mindformers/models/vit/vit.py:163
    def construct_without_pool(self, image, mask=None):
    ^
# In file /disk1/fail/scripts/mf_parallel0/mindformers/models/vit/vit.py:174
        for block in self.blocks:
                  ^
# In file
/disk1/fail/scripts/mf_parallel0/mindformers/models/vit/vit_modules.py:342
    def construct(self, x, input_mask, rel_pos_bias=None):
    ^
# In file
/disk1/fail/scripts/mf_parallel0/mindformers/models/vit/vit_modules.py:372
        mlp_logit = self.output(output_x)
                  ^
# In file
/disk1/fail/scripts/mf_parallel0/mindformers/models/vit/vit_modules.py:452
    def construct(self, x):
    ^
# In file
/disk1/fail/scripts/mf_parallel0/mindformers/models/vit/vit_modules.py:456
        hidden = self.mapping(x)
                 ^
# In file
/disk1/fail/scripts/mf_parallel0/mindformers/models/vit/vit_modules.py:198
    def construct(self, x):
    ^
# In file
/disk1/fail/scripts/mf_parallel0/mindformers/models/vit/vit_modules.py:201
        x = P.Reshape()(x, (-1, self.in_channels))
```

3. 根因分析

从报错中我们知道方法需要的参数是 636，但是我们提供的是 635，少一个。再结合上面的代码片段我们知道，调用的地方给了一个输入，在定义的地方我们定义了两个输入(image 和 text_input_ids)。所以会这里会报参数数目对不上的问题。

4. 解决方案

需改如下

```
# shard model and load sharded ckpt
warm_up_model = Model(model)
warm_up_model.infer_predict_layout(ms.Tensor(np.ones(shape=(1, 3, 336,336)),
ms.float32),ms.Tensor(np.ones(shape=(1, config.seq_length)), ms.int32))
```

这个报错下面的调用栈其实作用不大，当出现参数数目对不上的报错时，我们应该首先去排查给网络的输入参数是否和定义的一致。

5.13 MindSpore 数据并行报错 Call GE RunGraphWithStreamAsync Failed，EL0004: Failed to allocate memory.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.3.0

执行模式： 静态图

2. 报错信息

单机 8 卡 7B 参数量大小的模型，数据并行，报以下错误：

```
[ERROR] GE_ADPT(2032,fffcc57fd1e0,python):2024-01-26-18:16:34.140.365
[mindspore/ccsrc/transform/graph_ir/graph_runner.cc:352] RunGraphWithStreamAsync]
Call GE RunGraphWithStreamAsync Failed, ret is: 4294967295
2024-01-26 18:16:34,262 -
mindformers[mindformers/tools/cloud_adapter/cloud_monitor.py:43] - ERROR -
Traceback (most recent call last):
  File "/home/ma-user/modelarts/user-job-
dir/mindformers/mindformers/tools/cloud_adapter/cloud_monitor.py", line 34, in
wrapper
    result = run_func(*args, **kwargs)
  File "/home/ma-user/modelarts/user-job-dir/mindformers/run_mindformer.py", line
130, in main
    trainer.train(config, is_full_config=True)
  File "/home/ma-user/modelarts/user-job-
dir/mindformers/mindformers/trainer/causal_language_modeling/causal_language_modeli
ng.py", line 104, in train
    **kwargs)
  File "/home/ma-user/modelarts/user-job-
dir/mindformers/mindformers/trainer/base_trainer.py", line 738, in training_process
    initial_epoch=config.runner_config.initial_epoch)
  File "/home/ma-user/anaconda/lib/python3.7/site-
packages/mindspore/train/model.py", line 1073, in train
    initial_epoch=initial_epoch)
  File "/home/ma-user/anaconda/lib/python3.7/site-
packages/mindspore/train/model.py", line 114, in wrapper
    func(self, *args, **kwargs)
  File "/home/ma-user/anaconda/lib/python3.7/site-
packages/mindspore/train/model.py", line 624, in _train
    cb_params, sink_size, initial_epoch, valid_infos)
  File "/home/ma-user/anaconda/lib/python3.7/site-
packages/mindspore/train/model.py", line 708, in _train_dataset_sink_process
    outputs = train_network(*inputs)
  File "/home/ma-user/anaconda/lib/python3.7/site-packages/mindspore/nn/cell.py",
line 680, in __call__
    out = self.compile_and_run(*args, **kwargs)
  File "/home/ma-user/anaconda/lib/python3.7/site-packages/mindspore/nn/cell.py",
line 1023, in compile_and_run
    return _cell_graph_executor(self, *new_args, phase=self.phase)
```

```
  File  "/home/ma-user/anaconda/lib/python3.7/site-packages/mindspore/common/api.py",
line 1589, in __call__
    return self.run(obj, *args, phase=phase)
  File  "/home/ma-user/anaconda/lib/python3.7/site-packages/mindspore/common/api.py",
line 1628, in run
    return self._exec_pip(obj, *args, phase=phase_real)
  File  "/home/ma-user/anaconda/lib/python3.7/site-packages/mindspore/common/api.py",
line 121, in wrapper
    results = fn(*arg, **kwargs)
  File  "/home/ma-user/anaconda/lib/python3.7/site-packages/mindspore/common/api.py",
line 1608, in _exec_pip
    return self._graph_executor(args, phase)
RuntimeError: Exec graph failed
----------------------------------------------------
- Ascend Error Message:
----------------------------------------------------
EL0004: Failed to allocate memory.
       Possible Cause: Available memory is insufficient.
       Solution: Close applications not in use.
       TraceBack (most recent call last):
       Transport init error. Reason: [Create][DestLink]Create Dest error! creakLink
para:rank[7]-localUserrank[7]-localIpAddr[192.168.66.9], dst_rank[3]-
remoteUserrank[3]-remote_ip_addr[192.168.66.9]
```

根据以上信息的提示是： Failed to allocate memory. 大部分用户会以为是 bs 过大导致显存不足，但是当 bs 设置为 1 时依然报以上错误。

3. 根因分析

当将 bs 设置到最低的时候，还是显存不足，这时我们应该考虑到是否是因为系统给 hccl 分配的显存不足导致挂掉，我们可以尝试减小 max_device_memory 参数，显存不足的问题解决。

4. 解决方案

减小 max_device_memory 参数值。

5.14 MindSpore 分布式 8 节点报错 Call GE RunGraphWithStreamAsync Failed, ret is: 4294967295

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.0

执行模式： 不限

2. 报错信息

针对分布式集群训练，有的卡报错信息如下。

```
[ERROR] GE_ADPT(2245,fffa1ffff1e0,python):2024-01-03-13:56:51.823.242
[mindspore/ccsrc/transform/graph_ir/graph_runner.cc:352] RunGraphWithStreamAsync]
```

```
Call GE RunGraphWithStreamAsync Failed, ret is: 4294967295
EI0006: Getting socket times out. Reason: Remote Rank did not send the data in time.
Please check the reason for the rank being stuck
       Solution: 1. Check the rank service processes with other errors or no errors
in the cluster.2. If this error is reported for all NPUs, check whether the time
difference between the earliest and latest errors is greater than the connect
timeout interval (120s by default). If so, adjust the timeout interval by using the
HCCL_CONNECT_TIMEOUT environment variable.3. Check the connectivity of the
communication link between nodes. (For details, see the TLS command and HCCN
connectivity check examples.). For details:https://www.hiascend.com/document
       TraceBack (most recent call last):
       Call ops_kernel_info_store LoadTask
fail[FUNC:Distribute][FILE:hccl_task_info.cc][LINE:323]
       Call ops_kernel_info_store unloadTask
fail[FUNC:Release][FILE:hccl_task_info.cc][LINE:665]
       GraphManager RunGrapWithStreamhAsync failed,session id = 0, graph id = 2,
stream =
0xaaab1c1e9620.[FUNC:RunGraphWithStreamAsync][FILE:inner_session.cc][LINE:517]
       [Run][Graph]Run graph with stream asyn failed, error code = 1343225860,
session id = 0,graph id = 2, stream =
0xaaab1c1e9620.[FUNC:RunGraphWithStreamAsync][FILE:ge_api.cc][LINE:826]
```

以上报错的原因在报错中也给了几种排查方向，但是大概率是因为其中有一张卡挂掉导致建立通信失败。所以我们需要找到最新挂掉卡的那张卡的报错信息。本次业务中最先报错的是 device-5 信息如下:

```
[ERROR] GE_ADPT(2239,fffaa4ff91e0,python):2024-01-03-12:57:12.973.208
[mindspore/ccsrc/transform/graph_ir/graph_runner.cc:352] RunGraphWithStreamAsync]
Call GE RunGraphWithStreamAsync Failed, ret is: 4294967295
 File "/home/ma-user/anaconda3/envs/MindSpore/lib/python3.9/site-
packages/mindspore/train/model.py", line 623, in _train
    self._train_dataset_sink_process(epoch, train_dataset, list_callback,
  File "/home/ma-user/anaconda3/envs/MindSpore/lib/python3.9/site-
packages/mindspore/train/model.py", line 708, in _train_dataset_sink_process
    outputs = train_network(*inputs)
  File "/home/ma-user/anaconda3/envs/MindSpore/lib/python3.9/site-
packages/mindspore/nn/cell.py", line 680, in __call__
    out = self.compile_and_run(*args, **kwargs)
  File "/home/ma-user/anaconda3/envs/MindSpore/lib/python3.9/site-
packages/mindspore/nn/cell.py", line 1023, in compile_and_run
    return _cell_graph_executor(self, *new_args, phase=self.phase)
  File "/home/ma-user/anaconda3/envs/MindSpore/lib/python3.9/site-
packages/mindspore/common/api.py", line 1589, in __call__
    return self.run(obj, *args, phase=phase)
  File "/home/ma-user/anaconda3/envs/MindSpore/lib/python3.9/site-
packages/mindspore/common/api.py", line 1628, in run
    return self._exec_pip(obj, *args, phase=phase_real)
  File "/home/ma-user/anaconda3/envs/MindSpore/lib/python3.9/site-
packages/mindspore/common/api.py", line 121, in wrapper
    results = fn(*arg, **kwargs)
  File "/home/ma-user/anaconda3/envs/MindSpore/lib/python3.9/site-
packages/mindspore/common/api.py", line 1608, in _exec_pip
    return self._graph_executor(args, phase)
RuntimeError: Exec graph failed
----------------------------------------------------
- Ascend Error Message:
```

```
----------------------------------------------------
E19999: Inner Error!
E19999  Call ops_kernel_info_store LoadTask
fail[FUNC:Distribute][FILE:hccl_task_info.cc][LINE:323]
       TraceBack (most recent call last):
       Call ops_kernel_info_store unloadTask
fail[FUNC:Release][FILE:hccl_task_info.cc][LINE:665]
       GraphManager RunGrapWithStreamhAsync failed,session id = 0, graph id = 2,
stream =
0xaaaad7c58570.[FUNC:RunGraphWithStreamAsync][FILE:inner_session.cc][LINE:517]
       [Run][Graph]Run graph with stream asyn failed, error code = 1343225860,
session id = 0,graph id = 2, stream =
0xaaaad7c58570.[FUNC:RunGraphWithStreamAsync][FILE:ge_api.cc][LINE:826]
```

通过两个报错信息对比我们可以发现首先报错的时间点是这个 2024-01-03-12:57:12.973.208 我们可以记下这个时间。

当前业务的场景是 2 节点和 4 节点能够正常训练，8 节点挂死无法拉起训练，报以下错误。

```
call GE RunGraphWithStreamAsync Failed, ret is: 4294967295
```

当前报错无法提供任何有效信息，需要结合 plog 和 device 日志分析。

3. 根因分析

按照上面流程提到的我们需要结合 plog 和 device 日志进一步分析，我们在日志目录下面搜索这个 2024-01-03-12:57*时间点如下：

```
执行命令：grep -rani ERROR | grep 2024-01-03-12:57*
plog/plog-2239_20240103120735388.log:789:[ERROR] HCCP(2239,python):2024-01-03-
12:57:09.479.063 [ra_hdc.c:242]tid:103358,hdc_send_recv_pkt_recv(242) :
[recv][hdc_send_recv_pkt]HDC recv msg err ret(25)
plog/plog-2239_20240103120735388.log:790:[ERROR] HCCP(2239,python):2024-01-03-
12:57:09.479.094 [ra_hdc.c:308]tid:103358,hdc_send_recv_pkt(308) :
[send_recv][pkt]HDC pkt recv err ret(25) phy_id(0)
plog/plog-2239_20240103120735388.log:791:[ERROR] HCCP(2239,python):2024-01-03-
12:57:09.479.109 [ra_hdc.c:377]tid:103358,ra_hdc_process_msg(377) :
[process][ra_hdc_msg]hdc_send_recv_pkt opcode(22) failed ret(25) phy_id(0)
plog/plog-2239_20240103120735388.log:792:[ERROR] HCCP(2239,python):2024-01-03-
12:57:09.479.118 [ra_hdc.c:1713]tid:103358,ra_hdc_socket_white_list_op(1713) :
[op][ra_hdc_socket_white_list]ra hdc process msg failed ret(25) phy_id(0)
plog/plog-2239_20240103120735388.log:793:[ERROR] HCCL(2239,python):2024-01-03-
12:57:09.479.131
[adapter_hccp.cc:910][103358][Del][RaSocketWhiteList]errNo[0x000000000500000b] ra
white list del fail, return[328207].
plog/plog-2239_20240103120735388.log:794:[ERROR] HCCL(2239,python):2024-01-03-
12:57:09.479.138 [comm_base.cc:122][103358]call trace: hcclRet -> 11
plog/plog-2239_20240103120735388.log:795:[ERROR] HCCL(2239,python):2024-01-03-
12:57:09.479.145 [comm_base.cc:108][103358]call trace: hcclRet -> 11
plog/plog-2239_20240103120735388.log:796:[ERROR] HCCL(2239,python):2024-01-03-
12:57:09.479.150 [comm_base.cc:217][103358]call trace: hcclRet -> 11
plog/plog-2239_20240103120735388.log:797:[ERROR] HCCL(2239,python):2024-01-03-
12:57:09.479.156 [comm_base.cc:82][103358]call trace: hcclRet -> 11
plog/plog-2239_20240103120735388.log:798:[ERROR] HCCL(2239,python):2024-01-03-
12:57:09.479.162 [comm_factory.cc:776][103358][create][CommHD]comm halving doubling
```

```
array[0] init failed
.....................日志太多仅展示部分内容
```

从上面日志可以看到首报错日志是 HCCP 组件报错内容是 [ra_hdc.c:242]tid:103358,hdc_send_recv_pkt_recv(242) : [recv][hdc_send_recv_pkt]HDC recv msg err ret(25)

此类报错主要是因为 device 侧因为异常，先释放了 HDC 连接（message 日志中有 HDC 组件相应 info 级别日志提示：Server destroy success），导致 HCCP host 侧通过 hdc 通道下发 opcode 报错-25（session has been closed，表示对端即 device session 被关闭）。

问题根因：device 因为 OOM 提前释放了资源，导致 HCCP host 侧通过 hdc 下发 opcode 报错 25,(session has been closed,表示对端即 device seesion 被关闭)，其实是系统申请 device 内存不够所以才会 OOM，我们结合训练脚本中有设置设备可用的最大内存的配置 max_device_memory，此配置设置太大导致系统申请不到 device 内存导致挂掉。

2 节点和 4 节点没有出现这个报错是因为 2 节点和 4 节点的 dp 分别是 2 和 4，也就是 global batch size 是 2 和 4。

而到 8 节点时 global batch size 是 8。

4. 解决方案

尝试将 max_device_memory 调小以给系统预留足够显存。

5.15 MindFormers 进行单机八卡调用时报错 No parameter is entered. Notice that the program will run on default 8 cards.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.13

执行模式： 不限

2. 报错信息

2.1 问题描述

实执行如下命令出现报错。

```
(base) [root@bms910 mindformers]# bash scripts/msrun_launcher.sh "run_mindformer.py
--config {CONFIG_PATH} --run_mode {train/finetune/eval/predict}"
```

2.2 报错信息

```
No parameter is entered. Notice that the program will run on default 8 cards.
Running Command: msrun --worker_num=8   --local_worker_num=8   --master_port=8118
--log_dir=output/msrun_log   --join=False   --cluster_time_out=600
run_mindformer.py --config {CONFIG_PATH} --run_mode {train/finetune/eval/predict}
Please check log files in output/msrun_log
```

3. 根因分析

执行命令有误，查看 mindforers 的文档。

```
# 1. 单机多卡快速启动方式，默认 8 卡启动
bash scripts/msrun_launcher.sh "run_mindformer.py \
  --config {CONFIG_PATH} \
  --run_mode {train/finetune/eval/predict}"

# 2. 单机多卡快速启动方式，仅设置使用卡数即可
bash scripts/msrun_launcher.sh "run_mindformer.py \
  --config {CONFIG_PATH} \
  --run_mode {train/finetune/eval/predict}" WORKER_NUM

# 3. 单机多卡自定义启动方式
bash scripts/msrun_launcher.sh "run_mindformer.py \
  --config {CONFIG_PATH} \
  --run_mode {train/finetune/eval/predict}" \
  WORKER_NUM MASTER_PORT LOG_DIR JOIN CLUSTER_TIME_OUT
```

使用如下：

```
# 单机多卡快速启动方式，默认 8 卡启动
bash scripts/msrun_launcher.sh "run_mindformer.py \
  --config path/to/xxx.yaml \
  --run_mode finetune"

# 单机多卡快速启动方式
bash scripts/msrun_launcher.sh "run_mindformer.py \
  --config path/to/xxx.yaml \
  --run_mode finetune" 8

# 单机多卡自定义启动方式
bash scripts/msrun_launcher.sh "run_mindformer.py \
  --config path/to/xxx.yaml \
  --run_mode finetune" \
  8 8118 output/msrun_log False 300
```

4. 解决方案

修改命令，给出正确的执行命令。如果是训练 llama 模型则是

```
bash scripts/msrun_launcher.sh "run_mindformer.py \   --config
configs/llama/run_llama_7b.yaml \  --run_mode train"
```

5.16 MindSpore 分布式并行报错 The strategy is XXX, shape XXX cannot be divisible by strategy value XXX

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.1

执行模式： 静态图

2. 报错信息

2.1 报错信息

```
[ERROR] PARALLEL(371,ffff9fe22bf0,python):2023-11-10-09:29:03.578.320
[mindspore/ccsrc/frontend/parallel/ops info/operator info.cc:180]
CheckStrategyValue] AddInfo2323: The strategy is ((128, 1, 1), (128, 1, 1)), shape
32 cannot be divisible by strategy value 128
[ERROR] PARALLEL(371,ffff9fe22bf0,python):2023-11-10-09:29:03.578.372
[mindspore/ccsrc/frontend/parallel/ops info/operator info.cc:916]
InitForCostModelWithAutoRepeatCalc] AddInfo2323: CheckStrategy failed.
[ERROR] PARALLEL(371,ffff9fe22bf0,python):2023-11-10-09:29:03.578.384
[mindspore/ccsrc/frontend/parallel/ops info/operator info.cc:880] Init]
AddInfo2323 : Init failed.
[CRITICAL] PARALLEL(371,ffff9fe22bf0,python):2023-11-10-09:29:03.580.559
[mindspore/ccsrc/frontend/parallel/step parallel.cc:1953] ExtractStrategyAndInit]
Failure:operator Add init failed
The function call stack:
In file /opt/huawei/schedule-train/algorithm/src/pangu_alpha.py(664)/ output_states
= output_states + embedding_random/
In file /opt/huawei/schedule-train/algorithm/src/pangu_alpha.py(750)/ logits =
self.network(tokens,/
```

2.2 脚本信息

```
    def __init__(self, config):
        super(PanguAlphaModel, self).__init__()
        # Network head to get logits over vocabulary
        copied_parallel_config = copy.deepcopy(config.parallel_config)
        if copied_parallel_config.pipeline_stage > 1:
            copied_parallel_config.vocab_emb_dp = False
        self.head = PanGuHead(hidden_size=config.hidden_size,
                              parallel_config=copied_parallel_config)
        self.head.pipeline_stage = config.parallel_config.pipeline_stage - 1
        self.backbone = PanguAlpha_Model(config)

self.backbone.embedding.word_embedding.embedding_table.add_pipeline_stage(self.head
.pipeline_stage)

    def construct(self, input_ids, input_position, attention_mask,
                  init_reset=True, batch_valid_length=None):
        output_states, word_table = self.backbone(input_ids, input_position,
attention_mask,
                                                  init_reset, batch_valid_length)
        # 省略 embedding_random 的定义
        output_states = output_states + embedding_random
        logits = self.head(output_states, word_table)
        return logits
```

3. 根因分析

模型在 16 节点 128 卡上跑，model_parallel=8, data_parallel=16 根据调用栈，报错行为

```
output_states = output_states + embedding_random
```

output_states 和 embedding_random 的 shape 均为[32, 4096, 5120]这个加法没有配置切分策略，默认的切分策略是((128, 1, 1), (128, 1, 1))，输入的第一维是 32，不能被切分成 128 份。

4. 解决方案

此处不涉及模型权重，应该按数据并行维度 data_parallel=16 进行切分。定义 Add 算子并配置正确的切分策略((dp, 1, 1), (dp, 1, 1))。

```
def __init__(self, config):
    super(PanguAlphaModel, self).__init__()
    # Network head to get logits over vocabulary
    copied_parallel_config = copy.deepcopy(config.parallel_config)
    if copied_parallel_config.pipeline_stage > 1:
        copied_parallel_config.vocab_emb_dp = False
    self.head = PanGuHead(hidden_size=config.hidden_size,
                          parallel_config=copied_parallel_config)
    self.head.pipeline_stage = config.parallel_config.pipeline_stage - 1
    self.backbone = PanguAlpha_Model(config)

self.backbone.embedding.word_embedding.embedding_table.add_pipeline_stage(self.head
.pipeline_stage)
    self.add = P.Add().shard((config.parallel_config.data_parallel, 1, 1),
(config.parallel_config.data_parallel, 1, 1))

def construct(self, input_ids, input_position, attention_mask,
              init_reset=True, batch_valid_length=None):
    output_states, word_table = self.backbone(input_ids, input_position,
attention_mask,
                                              init_reset, batch_valid_length)
    # 省略 embedding_random 的定义
    output_states = self.add(output_states, embedding_random)
    logits = self.head(output_states, word_table)
    return logits
```

5.17 MTP 使用多进程生成 mindrecord，报错 RuntimeError: Unexpected error. [Internal ERROR] Failed to write mindrecord meta files.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.10

执行模式：Graph

2. 报错信息

```
File "make mindrecord v11. pv",line 404, in Kmodule>
    results = pool. map(map_func, whole_file_path)
  File */home/ma-
user/anaconda3/envs/1indSpor e/1ib/python3. 7/multiprocessing/pool. py*, line268,
in map
    return self. _map_async (func, iterable, mapstakr, chunksize). get()
  File */home/ma-
user/anaconda3/envs/Mindspor e/lib/python3. 7/multiprocessing/pool. py", line657,
```

```
in get
    raise self._value
RuntimeError: Unexpectederror. [Internal ERROR.Failed to write mindrecord meta
files.
```

3. 根因分析

根据报错信息，找到报错的代码行，发现使用 pool 进程池操作，需要检查数据量是否大于 pool 池数。

根因：多个进程不能写同一个 mindrecord 文件；或多进程处理的数据不能是空

4. 解决方案

多进程中分别使用的数据集改成大小不同的数据集后，未发生此问题。

或单数据集时，不使用 pool 操作。

5.18 流水线并行报错 Reshape op can't be a border.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.10

执行模式：PyNative/ Graph

2. 报错信息

```
    File "/home/ma-user/anaconda3/envs/python-3.9/1ib/python3.9/site
packages/mindspore/train/model. py", line 1274, in build
        self. _init(train_dataset, valid_dataset, sink_size, epoch)
    File "/home/ma-user/anaconda3/envs/python-3.9/1ib/python3.9/site-
packages/mindspore/train/model. py", line 529, in _init
        train_network. compile (*inputs)
    File "/home/ma-user/anaconda3/envs/python-3.9/1ib/python3.9/site
packages/mindspore/nn/cel1. py", line 997, in compile
        _cel1_graph_executor. compile (self, phase-self. phase,
    File "/home/ma-user/anaconda3/envs/python-3.9/1ib/python3.9/site-
packages/mindspore/common/api. py", line 1547, in compile
        result = self._graph_executor. compile(obj, args, kwargs, phase,
self._use_vm mode0)
RuntimeError: Reshape op can't bea border. node:@12339_src_ model_mllm_vi
sual_bridge_ VisualBridge_construct_852: output {[0]: ValueNode<PrimitivePy>
Reshape, [1]: output, [2]: ValueNode<Val1eTuple> (1, 32, 10240)}
-----------------------------------------------------------
- C++ Call Stack: (For framework developers)
-----------------------------------------------------------
mindspore/ccsrc/frontend/parallel/pipeline_ transformer/pipeline_transformer.
cc:556 Create0pInfo
```

3. 根因分析

Reshape 算子不能作为流水线并行的 stage 的边界，注意要排查调用的算子里有没有使用 Reshape 算子，比如 nn.Dense 的最后一步使用了 Rehape 算子，它放在流水线边界也是不行的。

网络脚本

```
self.pangu_proj = init_dense(
    in_channels=encoder_con fig.hidden_size,
    out_channels hidden_size,
    activation=None,
    init_method=init_method_normal('mindspore', sigma 0.02),
    dtype=ms.float32,
    compute_type ms.float32,
    has_bias=True
)
```

nn.Dense()源代码：

```
def construct(self, x):
    x_shape = self.shape_op(x)
    check_dense_input_shape(x_shape, self.cls_name)
    if len(x_shape) != 2:
        x = self.reshape(x, (-1, x_shape[-1]))
    x = self.matmul(x, self.weight)
    if self.has_bias:
        x = self.bias_add(x, self.bias)
    if self.activation_flag:
        x = self.activation(x)
    if len(x_shape) != 2:
        out_shape = x_shape[:-1] ] (F.shape(x)[-1],)
        x = self.reshape(x, out_shape)
    return x
```

4. 解决方案

1. 使用 copy 方法，比如上面的例子可以返回 output.copy()。（copy 方法里其实是将原数据除以 1.0）。

2. 移动边界位置。

5.19 增加数据并行数之后模型占用显存增加

1. 报错信息

增加数据并行数，同时按比例增加卡数，理论上每张卡上的模型大小不变，占用显存也应该不变，但是这里增加数据并行数之后显存 OOM 了，使用 export MS_MEMORY_STATISTIC=1 环境变量查看显存分配情况发现显存占用有明显提升。

```
dp=2
max static mem:1125453824 bytes,
peak used:1124418048 bytes,

dp=1
max static mem:973373440 bytes,
peak used:973373440 bytes,
```

2. 解决方案

- 参考文档 MindSpore 静态图模式内存分析优化方法进行定位，对 dp=1 和 dp=2 的情况分别跑 dry run，并用脚本分析日志，得到每个内存块的生命周期、大小、传递关系等信息（blocks.csv）。
- 对比两个场景，发现 sigmoid 算子显存占用显著增加。分别保存 ir 图，发现 dp=2 时 sigmoid 算子处做了重排布。

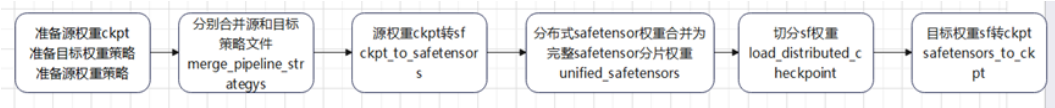
5.20 MindSpore 分布式 ckpt 权重 A 转换为其他策略的分布式权重 B

1. 使用场景

分布式策略改变后，训练所使用的卡数和分片策略都已改变，需要把源分布式权重转换为目标策略的分布式权重。

2. 解决方案

方法一：推荐转换流程



step1：源权重 A 准备

源权重存储要求：按照权重路径/rank_x/checkpoint_x.ckpt 存储，且每个 rank_x 下面只能有一个权重文件。 训练 yaml 配置 save_checkpoint_steps 时，训练会按照配置的值为间隔，进行权重保存，因此路径下会有多个权重

(可选)把需要的 ckpt 权重 A 挪到单独的目录中,保证每个 rank_x 下面只有一个 ckpt 文件

```
for i in {0..4095}; do mkdir -p /fs_ssd5_dpc_llm/rank_${i}; mv
/path/to/old_ckpt/rank_${i}/xxxx_rank_${i}_xxx.ckpt  /path/to/new_ckpt/rank_${i};
done
```

step2：分布式策略文件合并

需要把分布式权重 A、B 的策略文件分别合并

```
import mindspore as ms
ms.merge_pipeline_strategys(
src_strategy_dirs="源策略文件所在的目录",
dst_strategy_file="合并后的策略文件名")
```

src_strategy_dirs：A 或 B 的源策略文件,目录下面放了所有的策略文件,不需要按照 rank 存。

dst_strategy_file：合并后的策略文件,指定到策略文件名称

需要根据 https://gitee.com/mindspore/mindspore/pulls/76069/files 添加改动,否则合并效率很低

脚本：

```
import mindspore as ms
ms.merge_pipeline_strategys(
src_strategy_dirs="/xxx/ckpt_convert/142strategy/",
dst_strategy_file="/xxx/ckpt_convert/142merge.ckpt")
```

输入：

```
root@ /ckpt_convert# ll -r /  /ckpt_convert/142strategy/
total 116
-rw-------. 1 root root 53738 Nov 28 03:35 ckpt_strategy_rank_4.ckpt
-rw-------. 1 root root 53185 Nov 28 03:35 ckpt_strategy_rank_0.ckpt
drwxr-xr-x. 5 root root  4096 Nov 28 07:02 ../
drwxr-xr-x. 2 root root  4096 Nov 28 06:30 ./
root@ /ckpt_convert#
```

输出：

```
root@ /ckpt_convert# ll
total 160
drwxr-xr-x.  5 root root   4096 Nov 28 07:02 ./
drwxr-xr-x. 34 root root  12288 Nov 28 06:39 ../
-rw-r--r--.  1 root root 109249 Nov 28 06:30 142merge.ckpt
drwxr-xr-x.  2 root root   4096 Nov 28 06:30 142strategy/
drwxr-xr-x.  2 root root   4096 Nov 28 05:01 144strategy/
-rw-r--r--.  1 root root    383 Nov 28 06:48 1_to_n.py
-rw-r--r--.  1 root root    409 Nov 28 07:02 1_to_n_load_distributed_checkpoint.py
-rw-r--r--.  1 root root    206 Nov 27 14:17 ckpt_to_sf.py
-rw-r--r--.  1 root root   4435 Nov 27 06:45 hf_sf_to_cpkt.py
-rw-r--r--.  1 root root    327 Nov 27 07:39 load_distributed_checkpoint.py
-rw-r--r--.  1 root root    187 Nov 28 06:30 merge_pipeline_strategys.py
-rw-r--r--.  1 root root    197 Nov 27 08:17 sf_to_ckpt.py
drwxr-xr-x.  8 root root   4096 Nov 28 06:48 weight_o/
root@ /ckpt_convert#
```

s

tep3：权重 ckpt 格式转换为 safetensors

把分布式的 ckpt 权重转换成分布式的 safetensors 权重

```
import mindspore as ms
ms.ckpt_to_safetensors(
file_path="ckpt 文件的目录",
save_path="转换为 safetensors 文件的目录",
processes_num="进程数")
```

file_path：原始权重。

分布式权重：每个 rank 下面一个 ckpt,指定到 rank 的上一层路径。

完整权重：指定到权重文件。

save_path：转换为 safetensors 文件的目录

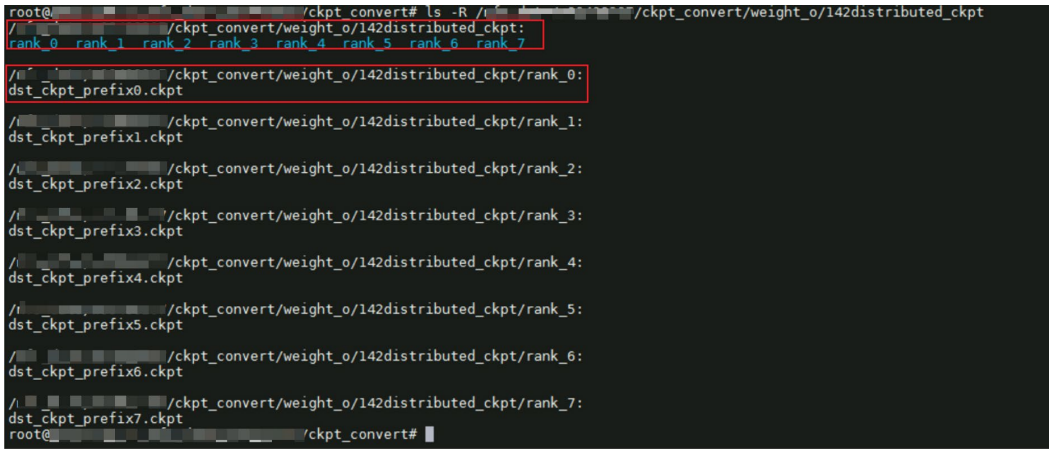
processes_num：整数,可取 8,16,32 等,根据自己机器内存、cpu 使用情况决定

样例：

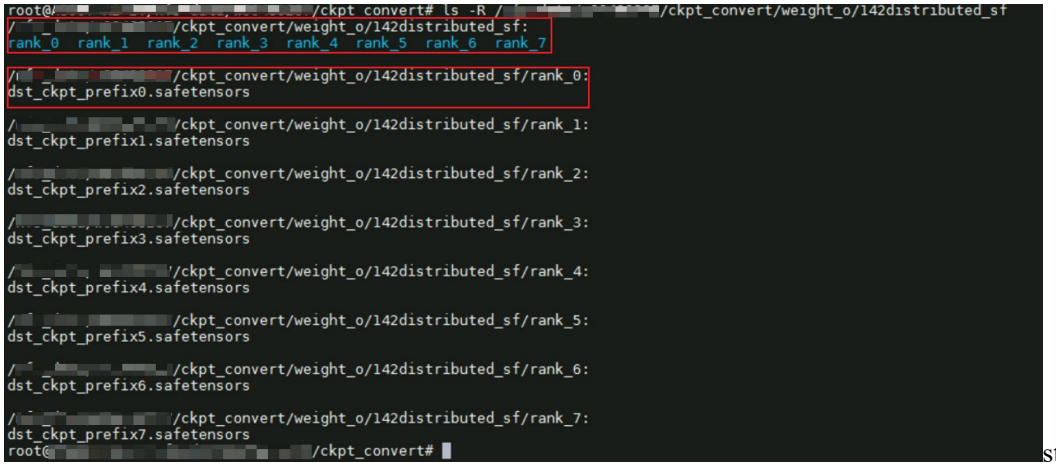
脚本：

```
import mindspore as ms
ms.ckpt_to_safetensors(
file_path="/xxx/ckpt_convert/weight_o/142distributed_ckpt",
save_path="/xxx/ckpt_convert/weight_o/142distributed_sf",
processes_num=8)
```

输入：



输出：



st

ep4：分布式 safetensors 权重合并为完整 safetensors 分片权重

```
import mindspore as ms
ms.unified_safetensors(
src_dir="safetensors 格式的文件目录",
src_strategy_file="权重 A 的切分策略(需合并)",
dst_dir="生成文件的目录")
```

src_dir: safetensors 分布式权重的目录

src_strategy_file：权重 A 的切分策略(需合并)

dst_dir：生成文件的目录

样例

脚本：

```
import mindspore as ms
ms.unified_safetensors(
src_dir="/xxx/ckpt_convert/weight_o/142distributed_sf",
src_strategy_file="/xxx/ckpt_convert/142merge.ckpt",
dst_dir="/xxx/ckpt_convert/weight_o/142merge_sf")
```

输入：合并的策略文件

```
root@ /ckpt_convert# ll
total 160
drwxr-xr-x.  5 root root   4096 Nov 28 07:02 ./
drwxr-xr-x. 34 root root  12288 Nov 28 06:39 ../
-rw-r--r--.  1 root root 109249 Nov 28 06:30 142merge.ckpt
drwxr-xr-x.  2 root root   4096 Nov 28 06:30 142strategy/
drwxr-xr-x.  2 root root   4096 Nov 28 05:01 144strategy/
-rw-r--r--.  1 root root    383 Nov 28 06:48 1_to_n.py
-rw-r--r--.  1 root root    409 Nov 28 07:02 1_to_n_load_distributed_checkpoint.py
-rw-r--r--.  1 root root    206 Nov 27 14:17 ckpt_to_sf.py
-rw-r--r--.  1 root root   4435 Nov 27 06:45 hf_sf_to_cpkt.py
-rw-r--r--.  1 root root    327 Nov 27 07:39 load_distributed_checkpoint.py
-rw-r--r--.  1 root root    187 Nov 28 06:30 merge_pipeline_strategys.py
-rw-r--r--.  1 root root    197 Nov 27 08:17 sf_to_ckpt.py
drwxr-xr-x.  8 root root   4096 Nov 28 06:48 weight_o/
root@ /ckpt_convert#
```

输入：分布式的 safetensors

```
root@ /ckpt_convert# ls -R /  /ckpt_convert/weight_o/142distributed_sf
/ /ckpt_convert/weight_o/142distributed_sf:
rank_0  rank_1  rank_2  rank_3  rank_4  rank_5  rank_6  rank_7

/ /ckpt_convert/weight_o/142distributed_sf/rank_0:
dst_ckpt_prefix0.safetensors

/ /ckpt_convert/weight_o/142distributed_sf/rank_1:
dst_ckpt_prefix1.safetensors

/ /ckpt_convert/weight_o/142distributed_sf/rank_2:
dst_ckpt_prefix2.safetensors

/ /ckpt_convert/weight_o/142distributed_sf/rank_3:
dst_ckpt_prefix3.safetensors

/ /ckpt_convert/weight_o/142distributed_sf/rank_4:
dst_ckpt_prefix4.safetensors

/ /ckpt_convert/weight_o/142distributed_sf/rank_5:
dst_ckpt_prefix5.safetensors

/ /ckpt_convert/weight_o/142distributed_sf/rank_6:
dst_ckpt_prefix6.safetensors

/ /ckpt_convert/weight_o/142distributed_sf/rank_7:
dst_ckpt_prefix7.safetensors
root@ /ckpt_convert#
```

输出：

```
root@ /ckpt_convert# ls -l /  /ckpt_convert/weight_o/142merge_sf -h
total 13G
-rw-------. 1 root root  12K Nov 28 14:59 hyper_param.safetensors
-rw-------. 1 root root  59K Nov 28 14:59 param_name_map.json
-rw-------. 1 root root 2.4G Nov 28 15:01 part0.safetensors
-rw-------. 1 root root 2.2G Nov 28 15:01 part1.safetensors
-rw-------. 1 root root 3.1G Nov 28 15:01 part2.safetensors
-rw-------. 1 root root 2.5G Nov 28 15:01 part3.safetensors
-rw-------. 1 root root 2.5G Nov 28 15:01 part4.safetensors
root@ /ckpt_convert#
```

s

tep5：切分 safetensors 权重

通过 load_distributed_checkpoint 可以将完整分片的 safetensors 权重转换为

1)完整 ckpt 2)分布式 ckpt 3)完整 safetensors 4）分布式 safetensors

备注：输入：格式 safetensors　输出：2.4.0 版本输出格式只能是 safetensors，2.4.0 之后可以是 ckpt 或 safetensors，默认'safetensors'

样例：8 卡 dp1mp4pp2 转 16 卡 dp1mp4pp4

脚本：

```
import mindspore as ms
ms.load_distributed_checkpoint(
network=None,
checkpoint_filenames=None,
predict_strategy="/xxx/ckpt_convert/144merge.ckpt",
format='safetensors',
output_format='safetensors',
unified_safetensors_dir="/xxx/ckpt_convert/weight_o/142merge_sf",
dst_safetensors_dir="/xxx/ckpt_convert/weight_o/144distributed_sf" )
```

输入：

```
[root@A    2 ~]# ll /ckpt_convert/144merge.ckpt
-rw-r--r--. 1 root root 107K Nov 28 15:04 /ckpt_convert/144merge.ckpt
[root@     ~]# ll /ckpt_convert/weight_o/142merge_sf
total 13G
-rw-------. 1 root root  12K Nov 28 14:59 hyper_param.safetensors
-rw-------. 1 root root  59K Nov 28 14:59 param_name_map.json
-rw-------. 1 root root 2.4G Nov 28 15:01 part0.safetensors
-rw-------. 1 root root 2.2G Nov 28 15:01 part1.safetensors
-rw-------. 1 root root 3.1G Nov 28 15:01 part2.safetensors
-rw-------. 1 root root 2.5G Nov 28 15:01 part3.safetensors
-rw-------. 1 root root 2.5G Nov 28 15:01 part4.safetensors
[root@     ~]#
```

输出分布式权重：

```
# ll 7/ckpt_convert/weight_o/144distributed_sf -R
/ckpt_convert/weight_o/144distributed_sf:
total 64K
drwx------. 2 root root 4.0K Dec 31 03:36 rank_0
drwx------. 2 root root 4.0K Dec 31 03:36 rank_1
drwx------. 2 root root 4.0K Dec 31 03:36 rank_10
drwx------. 2 root root 4.0K Dec 31 03:36 rank_11
drwx------. 2 root root 4.0K Dec 31 03:36 rank_12
drwx------. 2 root root 4.0K Dec 31 03:36 rank_13
drwx------. 2 root root 4.0K Dec 31 03:36 rank_14
drwx------. 2 root root 4.0K Dec 31 03:36 rank_15
drwx------. 2 root root 4.0K Dec 31 03:36 rank_2
drwx------. 2 root root 4.0K Dec 31 03:36 rank_3
drwx------. 2 root root 4.0K Dec 31 03:36 rank_4
drwx------. 2 root root 4.0K Dec 31 03:36 rank_5
drwx------. 2 root root 4.0K Dec 31 03:36 rank_6
drwx------. 2 root root 4.0K Dec 31 03:36 rank_7
drwx------. 2 root root 4.0K Dec 31 03:36 rank_8
drwx------. 2 root root 4.0K Dec 31 03:36 rank_9

/ckpt_convert/weight_o/144distributed_sf/rank_0:
total 835M
-r--------. 1 root root 835M Dec 31 03:36 net.safetensors

/ckpt_convert/weight_o/144distributed_sf/rank_1:
total 835M
-r--------. 1 root root 835M Dec 31 03:36 net.safetensors

/ckpt_convert/weight_o/144distributed_sf/rank_10:
total 773M
-r--------. 1 root root 773M Dec 31 03:36 net.safetensors

/ckpt_convert/weight_o/144distributed_sf/rank_11:
total 773M
-r--------. 1 root root 773M Dec 31 03:36 net.safetensors

/ckpt_convert/weight_o/144distributed_sf/rank_12:
total 835M
-r--------. 1 root root 835M Dec 31 03:36 net.safetensors

/ckpt_convert/weight_o/144distributed_sf/rank_13:
total 835M
-r--------. 1 root root 835M Dec 31 03:36 net.safetensors

/ckpt_convert/weight_o/144distributed_sf/rank_14:
total 835M
-r--------. 1 root root 835M Dec 31 03:36 net.safetensors
```

s

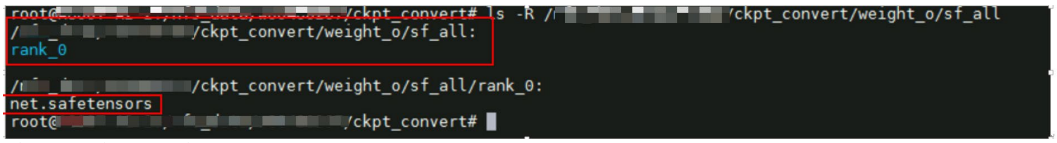
tep6: (2.4.0 必须) safetensors 转 ckpt

```
import mindspore as ms
ms.safetensors_to_ckpt(
file_path="safetensors 权重文件或目录 ",
save_path="转换为 ckpt 文件的目录",
processes_num="进程数")
```

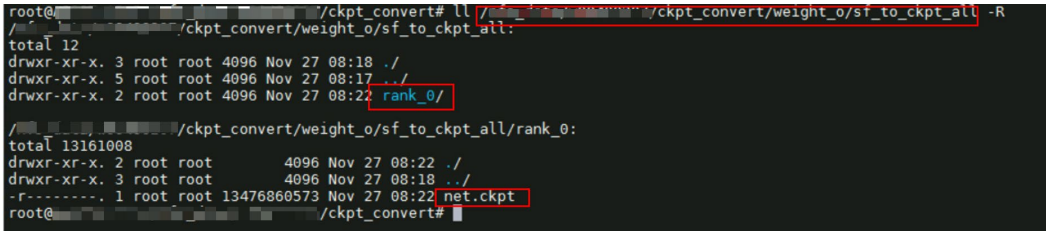
样例：

```
import mindspore as ms
ms.safetensors_to_ckpt(
file_path="/xxx/ckpt_convert/weight_o/sf_all",
save_path="/xxx/ckpt_convert/weight_o/sf_to_ckpt_all",
processes_num= 8)
```

输入：



输出完整权重截图：



方法二：小权重转换使用 transform_checkpoints

```
import mindspore as ms
ms.transform_checkpoints (src_checkpoints_dir, dst_checkpoints_dir, ckpt_prefix,
src_strategy_file=None, dst_strategy_file=None, process_num=1, output_format='ckpt')
print("Transform ckpt DONE", flush=True)
```

src_checkpoints_dir:权重 A 的路径,分布式权重指定到 rank 的上一层路径,完整权重指定到文件名

dst_checkpoints_dir:生成权重 B 的路径

ckpt_prefix：生成权重 B 文件名的前缀

src_strategy_file：权重 A 合并后的策略文件,对于完整权重指定 None

dst_strategy_file：权重 B 合并后的策略文件,对于完整权重指定 None

process_num：线程格数,根据自己机器 cpu、内存使用情况绝对

output_format："ckpt" 或"safetensors"

5.21 流水线并行没开 Cell 共享导致编译时间很长

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.10

执行模式： 不限

Python 版本: 3.8.15

操作系统平台: linux

2. 报错信息

使用了流水线并行，模型编译时间很长(几个小时)，或者因为编译时间太长误以为是卡住。

3. 根因分析

在大模型场景中，编译耗时问题尤为突出，一是大模型的模型结构层次深，节点数多；二是大模型在训练时，由于启用 pipeline 并行，导致模型规模和节点数进一步加大，如果原来图的规模是 O，那开启 pipeline 并行，单节点图的规模变为(O/X)*Y，其中 X 为 pipeline 的 stage 数量，Y 为 micro batch 的数量。以盘古 13B 网络为例，计算图中计算节点数量达到 13.5 万个，单次编译时长可接近 3 小时。

4. 解决方案

使用 lazy_inline 装饰器优化编译性能

MindSpore 2.2 版本前：控制是否使能 lazy_inline 是通过@lazy_inline 加上 MS_DEV_CELL_REUSE 这个环境变量。

MindSpore 2.2 及以后版本：控制 lazy_inline 是否使能只需通过@lazy_inline。

如果使用的是 mindformers：mindformers 上为了做到对不同版本的 ms，通过同一个环境变量统一控制是否开启 lazy_inline，搞了个环境变量 ENABLE_CELL_REUSE 来控制是否使能。

5.22 多机训练报错：import torch_npu._C ImportError: libascend_hal.so: cannot open shared object file: No such file or directory

1. 报错信息

执行多机训练报错，单机执行相同命令可正常启动训练

脚本信息：

```
import torch_npu._C
ImportError: libascend_hal.so: cannot open shared object file: No such file or
directory
```

报错信息

```
Traceback (most recent call last):
File
"/work/share/lcy/audio/CosyVoice/examples/libritts/cosyvoice/cosyvoice/bin/train.py
", line 15, in <module>
  import torch_npu
File "/usr/local/lib/python3.10/dist-packages/torch_npu/__init__.py", line 17, in
<module>
  import torch_npu.npu
File "/usr/local/lib/python3.10/dist-packages/torch_npu/npu/__init__.py", line 122,
in <module>
  from torch_npu.utils import should_print_warning
File "/usr/local/lib/python3.10/dist-packages/torch_npu/utils/__init__.py", line 2,
in <module>
  from ._module import _apply_module_patch
File "/usr/local/lib/python3.10/dist-packages/torch_npu/utils/_module.py", line 26,
in <module>
  from torch_npu.npu.amp.autocast_mode import autocast
File "/usr/local/lib/python3.10/dist-packages/torch_npu/npu/amp/__init__.py", line
```

```
7, in <module>
  from .sharded_grad_scaler import ShardedGradScaler
File "/usr/local/lib/python3.10/dist-
packages/torch_npu/npu/amp/sharded_grad_scaler.py", line 9, in <module>
  from torch_npu.npu.utils import npu_check_overflow
File "/usr/local/lib/python3.10/dist-packages/torch_npu/npu/utils.py", line 12, in
<module>
  import torch_npu._C
ImportError: libascend_hal.so: cannot open shared object file: No such file or
directory
```

2. 根因分析

多机执行命令如下：

```
sudo docker exec {$docker_id} /bin/bash -c 'source /usr/local/Ascend/ascend-
toolkit/set_env.sh; pkill -9 python; pkill -9 torchrun; cd /xxx1; bash
train_multinode.sh llm /xxx2/hostfile 60009;'
```

拉起训练任务失败。

单机执行命令如下：

```
# 先执行 docker exec -it {$docker_id} bash 手动进入容器
bash train_multinode.sh llm /xxx2/hostfile 60009
```

拉起训练任务成功。

1）登录到物理机上，手动进入容器，执行训练命令，能拉起训练任务

2）登录到物理机上，不进入容器，在容器外通过 docker exec 执行命令，报错，但在命令中加上环境变量之后也不报错了

3）不登录物理机，通过 HJ 平台拉起任务，报错，加上环境变量还是报错

从报错日志看，找不到 libascend_hal.so 文件，通过 find / -name 'libascend_hal.so'命令找到该文件在/usr/local/Ascend/driver/lib64/driver 目录下，需要将该路径加到 $LD_LIBRARY_PATH 系统环境变量下。

```
export LD_LIBRARY_PATH=/usr/local/Ascend/driver/lib64/driver:$LD_LIBRARY_PATH
```

将环境变量加入到训练脚本中后，HJ 平台可正常拉起多机训练任务。

手动执行不报错，因为该环境变量在容器镜像中已加入到了/root/.bashrc 中，手动登录容器会自动加载。但在容器外执行或者通过 HJ 平台调用时，不会加载。

在 HJ 平台加上环境变量还是报错：因为 HJ 平台加环境变量时，source 的 setenv.sh 实际未执行成功（因为容器内没挂载对应的脚本）。

总结：碰到 libascend_hal.so 库文件找不到的问题时，检查环境变量 env | grep LD_LIBRARY_PATH ，查看 LD_LIBRARY_PATH 环境变量中是否包含 /usr/local/Ascend/driver/lib64/driver 路径，无论通过哪种方式加载，最终确保 LD_LIBRARY_PATH 环境变量中包含/usr/local/Ascend/driver/lib64/driver 路径即可。

LD_LIBRARY_PATH 环境变量未包含/usr/local/Ascend/driver/lib64/driver 路径。

4. 解决方案

将/usr/local/Ascend/driver/lib64/driver 路径添加到 LD_LIBRARY_PATH 环境变量中。

5.23 docker 执行报错：RuntimeError: Maybe you are trying to call 'mindspore.communication.init()' without using 'mpirun'

1. 问题现象

docker 内执行脚本报错：

```
RuntimeError: Maybe you are trying to call 'mindspore.communication.init()' without
using 'mpirun', which will make MindSpore load several environment variables and
check their validation. Please use 'mpirun' to launch this process to fix this
issue, or refer to this link if you want to run distributed training without using
'mpirun': https://www.mindspore.cn/tutorials/experts/zh-
CN/master/parallel/train_gpu.html
```

容器外执行 npu-smi info 可以看到卡未被占用。容器内执行 npu-smi info 报错

```
DrvMngGetConsoleLogLevel failed. (g_conLogLevel=3)
dcmi model initialized failed, because the device is used. ret is -8020
```

2. 问题根因

卡已经被其他 docker 占用，导致当前 docker 加载不上卡，导致报错，采用如下方式启动的 docker 是独占卡的，单张卡只能被单个 docker 加载，导致其他 docker 内看不到卡。

```
--device=/dev/davinci0 \
--device=/dev/davinci1 \
--device=/dev/davinci2 \
--device=/dev/davinci3 \
--device=/dev/davinci4 \
--device=/dev/davinci5 \
--device=/dev/davinci6 \
--device=/dev/davinci7 \
--device=/dev/davinci manager \
--device=/dev/devmm svm \
--device=/dev/hisi hdc \
```

3. 解决方案

使用这种方式加载的卡可以支持多 docker 共用，docker 内都可以看到卡的状态。

```
docker run -itd --net=host --cap-add=SYS_PTRACE -e ASCEND_VISIBLE_DEVICES=0-7
```

6 训练推理问题案例

6.1 MindSpore 报错：TypeError: Multiply values for specific argument: query_embeds

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.1

执行模式： 静态图

2. 报错信息

2.1 问题描述

使用 MindSpore 开发模型报以下错误：

```
TypeError: Multiply values for specific argument: query_embeds
```

2.2 报错信息

```
Traceback (most recent call last):
  File "/a.py", line 95, in <module>
    output = model(img_tensor)
  File "/root/miniconda3/envs/MindSpore2.1.1py39/lib/python3.9/site-
packages/mindspore/nn/cell.py", line 637, in __call__
    out = self.compile_and_run(*args, **kwargs)
  File "/root/miniconda3/envs/MindSpore2.1.1py39/lib/python3.9/site-
packages/mindspore/nn/cell.py", line 961, in compile_and_run
    self.compile(*args, **kwargs)
  File "/root/miniconda3/envs/MindSpore2.1.1py39/lib/python3.9/site-
packages/mindspore/nn/cell.py", line 938, in compile
    _cell_graph_executor.compile(self, phase=self.phase,
  File "/root/miniconda3/envs/MindSpore2.1.1py39/lib/python3.9/site-
packages/mindspore/common/api.py", line 1623, in compile
    result = self._graph_executor.compile(obj, args, kwargs, phase,
self._use_vm_mode())
TypeError: Multiply values for specific argument: query_embeds

----------------------------------------------------
- Ascend Warning Message:
----------------------------------------------------
W29999: [Init][LoadAscendCustomOppPath]
```

```
ASCEND_CUSTOM_OPP_PATH[/root/miniconda3/envs/MindSpore2.1.1py39/lib/python3.9/site-
packages/mindspore/run_check/../lib/plugin/ascend/custom_aicpu_ops] is not exist or
not abstract path.[FUNC:ResolveOpImplPath][FILE:configuration.cc][LINE:683]

----------------------------------------------------
- The Traceback of Net Construct Code:
----------------------------------------------------
The function call stack (See file '/rank_0/om/analyze_fail.ir' for more details.
Get instructions about `analyze_fail.ir` at
https://www.mindspore.cn/search?inputValue=analyze_fail.ir):
# 0 In file /a.py:75
       output = self.qformer(

----------------------------------------------------
- C++ Call Stack: (For framework developers)
----------------------------------------------------
mindspore/core/ir/func_graph_extends.cc:172 GenerateKwParams
```

2.3 相关脚本

```
    def construct(self, img_tensor: ms.Tensor):
        """
        img_tensor: [bs, c, h, w]
        """
        img_embeds = self.vmodel(img_tensor)
        img_atts = ms.Tensor(np.ones(img_embeds.shape[:-1]), dtype=ms.float32)
        output = self.qformer(
            query_embeds=self.query_tokens,
            encoder_hidden_states=img_embeds,
            encoder_attention_mask=img_atts
        )
        output = self.pangu_proj(output)
        return output
```

3. 根因分析

根据报错信息可以知道我们给 query_embeds 传了多个值，但是实际上是只需要一个即可，所以我们会首先看下传入的这个 self.query_tokens 是不是有问题，我们通过打印分析，self.query_tokens 是一个 Tensor，不存在多值的情况。

此时我们问题定位不仅仅只关注报错，可能是其他问题诱发这个给人误导性的报错，我们走读脚本发现在报错代码的上面有一句：

```
img_atts = ms.Tensor(np.ones(img_embeds.shape[:-1]), dtype=ms.float32)
```

此时猜测可能是由于在图模式下,Cell 的 construct 中引用了 Numpy。

4. 解决方案

我们将代码做修改，原代码：

```
img_atts = ms.Tensor(np.ones(img_embeds.shape[:-1]), dtype=ms.float32)
```

修改后：

```
img_atts = ms.ops.ones(img_embeds.shape[:-1], ms.float32)
```

再次执行，模型可以顺利跑通。

6.2 Tokenizer 指向报错 TypeError GPT2Tokenizer: __init__() missing 2 required positional arguments: 'vocab_file' and "merges_file'.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 报错信息

2.1 问题描述

执行 WizardCoderTokenizer.from_pretrained(…)命令时，没有指向 WizardCoderTokenizer，反而指向了 GPT2Tokenizer，导致加载错误。

```
Traceback (most recent call last):
File "/home/wizardcoder/1_wizardcoder-
mindformers/mindformers/tools/register/register.py", line 217, in get_instance
return obj_cls(**kwargs)
TypeError:_init_( ) missing 2 required positional arguments: 'vocab_file' and
'merges_file'
During handling of the above exception, another exception occurred:
Traceback (most recent call last):File "<stdin>", line 1, in<module>  File
"/home/wizardcoder/1_wizardcoder-mindformers/mindformers/models/base_tokenizer.py"  ,
line 2004, in from_pretrained  return build_tokenizer(class_name=class_name ,
**kwargs)  File "/home/wizardcoder/1_wizardcoder-
mindformers/mindformers/models/build_tokenizer.py", line 76, in build_tokenizer
return MindFormerRegister. get_instance(module_type , class_name, **kwargs)   File
"/home/wizardcoder/1_wizardcoder-
mindformers/mindformers/tools/register/register.py", line 219, in get_instance
raise tvnelel('{}. {} ".format(obj_cls._name_, e))    TypeError: GPT2Tokenizer:
init () missing 2 required positional arguments: 'vocab_file' and 'merges_file'
```

3. 根因分析

注册或者配置文件中可能指向了 GPT2Tokenizer，需要排查：

1） mindformer_book.py 中查看 tokenizer 是否注册

2） run_wizardcoder.yaml 文件中查看是否有指向错误

3） from_pretrained 引用的预训练模型文件中查看是否有指向错误

4. 解决方案

发现在 tokenizer_config.json 文件中引用了 GPT2Tokenizer，需要将其改为 WizardCoderTokenizer。

```
"tokenizer_class":"GPT2Tokenizer"
```

改为

```
"tokenizer_class":"WizardCoderTokenizer"
```

6.3 Tokenizer 文件缺失报错 TypeError:__init_() missing 2 required positional arguments: 'vocab_file' and 'merge_file'

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 报错信息

2.1 问题描述

加载 WizardCoderTokenizer.from_pretrained(…)时出现报错，发现缺少两个文件 vocab_file 和 merge_file。

```
Traceback (most recent call last):  File /home/wizardcoder/1_wizardcoder-
mindformers/mindformers/tools/register/register.py" , line 217, in get_instance
return obi cls(**kwaras)
TypeError:__init_() missing 2 required positional arguments: 'vocab_file' and
'merge_file'
```

3. 根因分析

在 tokenizer_config.json 文件新增两列，指定这两个文件路径

4. 解决方案

需在 tokenizer_config.json 文件中增加 vocab_file 和 merge_file 对应的文件路径。

```
"bos_token": "<lendoftext|>",
"clean_up_tokenization_spaces": true,
"eos token": "＜lendoftext|>",
"model_max_length": 2048,
"padding_side": "right",
"tokenizer_class": "WizardCoderTokenizer",
"unk_token": "<lendoftext|>",
"vocab size": 49152,
"vocab_file": "/xxxxxx/wizardcoder/vocab.json",
"merge_file":  '/xxxxxx/wizardcoder/merges.txt
```

6.4 模型推理时验证 model.generate()功能报错 RuntimeError A model class needs to define a `prepare inputs fordgeneration` method in order to use .generate()`

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 报错信息

2.1 问题描述

模型推理时验证 model.generate()功能报错。

```
Traceback (most recent call last):
File "predict dx.py". line 26, in <module>
result = model.generate(input_ids=inputs["input_ids"],
max_length=model.config.seq_length)
File "/home/wizardcoder/1_wiz ardcoder-
mindformers7mindformers/generation/text_generator.py", line, in generate**model
kwargs,
File "/home/wizardcoder/1 wizardcoder-
mindformers/mindformers/generation/text_generator.py", line 329, in _forward
input ids. **model kwargs
File. "/home/wizardcoder/1 wizardcoder-mindformers /mindformers/generation/text
generator.py", line 58, in prepare inputs for generation
"A model class needs to-define a `prepare inputs for aeneration*
RuntimeError A model class needs to define a `prepare inputs fordgeneration` method
in order to use .generate()`
```

3. 根因分析

分析代码可知（见 text_generator.py 第 48 行），如果需要使用 model.generate()功能，必须要重写 prepare_inputs_for_generation()方法。

```
class GeneratorMixin:
    """Generator For the nlp models"""
    def __init__(self):
        pass
    # pylint: disable=W0613
    def prepare_inputs_for_generation(self, input_ids, **kwargs):
        """
        prepare inputs for generation.
        A model class needs to define a `prepare_inputs_for_generation` method
        in orderto use `.generate()
        Raises:
            RuntimeError: Not implemented in model but call `.generate()`
        """
        raise RuntimeError(
            "A model class needs to define a `prepare_inputs_for_generation`"
            " method in order to use `.generate()`."
        )
```

4. 解决方案

需要在待迁移模型中实现这个功能方法，参照 GPT-2 的写法，wizardcoder 在 WizardCoderLMHeadModel 类中实现了这个功能。

```
def prepare_inputs_for_generation(self, input_ids, **kwargs):
    input_ids = {"input_ids":Tensor(input_ids, mstype.int32)}
    # input_ids = {"input_ids": input_ids}

    Return input_ids
```

6.5 MindSpore 模型推理报错：memory isn't enough and alloc failed, kernel name: kernel_graph_@ HostDSActor, alloc size: 8192B

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.0&CANN7.0

执行模式： 不限

2. 报错信息

2.1 问题描述

使用 MindSpore2.2.0&CANN7.0，wizardcoder 模型层数设为 4，主要目的是快速验证是否能正确推理，跑推理脚本时出现报错。

2.2 报错信息

```
[WARNING] GE_ ADPT(3981562,ffff8dda3af0,python):2023-10-26-15:56:10.535.470
[mindspore/ccsrc/transform/graph_ir/utils.cc:300] CheckAndGetGraphRunner] Can not
find init_subgraph.kernel_graph_0 sub graph, don't need data init subgraph in INFER
mode.
[WARNING] GE_ADPT(3981562, ffff8dda3af0,python) :2023-10-26- 15:56: 10.577.631
[mindspore/ccs rc/transform/graph_ ir/ut ils.cc:300] CheckAndGe tGraphRunner] Can
not find init _subgraph.496_1_wizardcode r WizardCoderLMHeadModel_construct 98 sub
graph, don't need data init subgraph in INFER mode.
[TRACE] GE(3981562,python):2023-10-26-15:56:10.579.360 [status:INIT]
[ge_api.cc :964]3984989 RunGraphAsync :start to run graph async, session_id: 0,
graph_id: 1,input size 1
Traceback (most recent call last):
   File "test_wizardcoder_generate .py", line 67, in <module>
      main(config_path=args.config, max_length=args.max_length,
use_past=args.use_past, batch_size=args.batch_size, device_id=args.device_id)
   File "test_wizardcoder_generate -py", line 47, in main
      output = model.generate(input ids=tokenizer(prompt) ["input ids"], use
past=use past, max length=max length)
   File "/home/wizardcoder/2 wizardcoder-mindformers-
1019/mindtormers/generation/text_generator.py", line 565, in generate
      **model kwargs,
   File "/home/wizardcoder/2 wizardcoder-mindformers-
1019/mindformers/generation/text generator.py",ine39 in fonward
      res = self(**model inputs) # pylint: disable=E1102
   File "/home/miniconda3/1 ib/python3.7/site-packages/mindspore/nn/cell.py", line
680, in _call
      out = self.compile and run(*args, **kwargs)
   File "/home/miniconda3/lib/python3.7/site-packages/mindspore/nn/cell.py", line
1023, in compile_and_run
      return cell graph executor(self, *new args, phase=self.phase)
   File "/home/miniconda3/lib/python3.77site-packages/minds pore/common/api.py",
line 1589, in _call_
      return self.run(obj, *args, phase=phase)
   File "/home/miniconda37lib/python3.7/site-packages/minds pore/common/api .py",
```

```
line 1628, in run
       return self._exec_pip(obj , *args, phase=phase real)
   File "/home/miniconda3/lib/python3.7/site-packages/minds pore/common/api.py",
line 121, in wrapper
       results = fn(*arg, **kwargs)
   File "/home/miniconda3/l ib/python3.7/site-packages/minds pore/common/api.py",
line 1608, in _exec_pip
       return self._graph_executor (args, phase)
RuntimeError:
   Memory not enough:
Device(id:7) memory isn't enough and alloc failed, kernel name: kernel_graph_0 Hos
tDSActor, alloc size: 8192B.
- C++ Call Stack: (For framework developers)
mindspore/ccsrc/ runtime/graph scheduler/graph scheduler.cc:679 Run
```

3. 解决方案

通过排查发现是设置了环境变量 “MS_DISABLE_REF_MODE=1”导致出错，unset 这一环境变量之后可以正常推理。

6.6 MindSpore 模型推理报错 TypeError: cell_reuse() takes 0 positional arguments but 1 was given

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.0

执行模式： 不限

2. 报错信息

2.1 问题描述

使用 MindSpore 进行模型推理时报错。

2.2 报错信息

```
ml#keyword-namespace-packages
   declare_namespace(pkg)
Traceback (most recent Tcall last):
   File "pipeline_predict.py", line 5, in <module>
       from wizardcoder import WizardCoderLMHeadModel
   File "/home/wizardcoder/2_ wizardcoder-mindformers-
1019/research/wizardcoder/wizardcoder.py", line 49, in <module>
       class WizardcoderLMHeadModel (BaseModel):
   File "/home/wizardcoder/2_wizardcoder-mindformers-1019/research/wizardcode
r/wizardcoder.py", line 59, in Wiza rdcoderLMHeadModel
       def __init__(self, config:WizardCoderConfig = None):
TypeError: cell_reuse() takes 0 positional arguments but 1 was given
```

3. 根因分析

上述报错意思是调用 cell_reuse()报错，这是由于 mindspore2.2.0 版本不支持 cell_reuse()这种写法。

4. 解决方案

由于 mindspore2.2.0 版本不支持 cell_reuse()这种写法，所以应该写成 cell_reuse，代码如下。

```
@MindFormerRegister.register(MindFormerModuleType. MODELS)
class WizardCoderLMHeadModel(BaseModel):
    r"""..."""
    @cell_reuse
    def __init__ (self, config: WizardCoderConfig = None):
        config = config if config is not None else WizardCoderConfig()
        super (WizardCoderLMHeadModel, self).__init__(config, auto_prefix=True)
        self.use_past = config.use_past
```

装饰器名称改为 cell_reuse 后如果还是报上面错误，这应该是因为本身环境安装了 mindformers 旧版本，代码可能调用了安装的旧版本 mindformers 的功能，导致出错。解决方法是添加调用路径，让 cell_reuse 调用我们当前的 mindformers dev 版本：

```
import os
import sys

sys.path.append(os.path.abspath("../.."))
sys.path.insert(0, os.getcwd().split('research')[0])

from mindspore import context
from mindformers.pipeline import pipeline
```

6.7 运行 wizardcoder 迁移代码报错 broken pipe

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 报错信息

2.1 问题描述

之前能够跑通的 wizardcoder 训练代码，在引入 mindspore.ops.Print()后再次运行无法跑通。

2.2 报错信息

```
uring handling of the above exception, another exception occurred:
    File "/home/liyejun/miniconda3/envs /test0823/lib/python3.7/multiproc
essing/connection.py", line 368, in _send
        n = write(seif. handle. buf)
rokenPipeError: [Errno 32] Broken pipe
```

```
raceback (most recet catl last)
rokenPipeError: [Errno 32] Broken pipe
```

3. 根因分析

使用日志打印功能来查看。命令如下：

```
export GLOG_v = 1: mindspore 日志输出等级设置，设为 2 后停止输出
export ASCEND_GLOBAL_LOG_LEVEL=1: CANN 相关日志等级
export ASCEND_SLOG_PRINT_TO_STDOUT=1: CANN 相关日志控制,0 不输出,1 打屏
```

找到出错的位置，查看前后信息，有一行信息是“type of scalar to print is not support”，推测是在调试的过程中使用了 mindspore 的 mindspore.ops.Print()，打印的对象不满足要求，导致报错。

4. 解决方案

在 wizardcoder.py 文件中添加了上述 print 功能，将其注释或者删除后可以跑通，具体位置在 WizardCoderLMHeadModel 类的__init__()和 construct()中。

```
        self.tile = P.Tile()
        self.concat = P.Concat(axis=-1)
        # self.print - = mindspore.ops.Print()
        def _add_eos_to_inputs(self, target_ids):...
              output_states, table = self.backbone(tokens, attention_mask)
        # self.print('------------------------')
        # self.print(output_states. shape)
        # self.print(table.shape)
        logits = self.head(output_states table)
```

分析官网对于 ops.Print()的介绍(参见：https://www.mindspore.cn/docs/zh-CN/r2.1/api_python/ops/mindspore.ops.Print.html#mindspore.ops.Print)。官网说明 ops.Print()的入参可以是 str、Tensor、int 等的 union 组合，只注释如下行代码，发现能够正确跑通代码且打印出对应的 shape 值。

```
        # self.print('------------------------')
        self.print(output_states. shape)
        self.print(table.shape)
        logits = self.head(output_states table
```

6.8 MindSpore Lite 模型加载报错 RuntimeError: build from file failed! Error is Common error code.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.1

执行模式： 不限

2. 报错信息

2.1 问题描述

使用 MindSpore Lite 增量推理，在加载模型时报错。

2.2 报错信息

```
[TRACE] GE(334540, python) :2023-10-13-18:54:37.246.214 [status:STOP]
[ge_api.cc :316]334540 GEInitializeImpl :GEInitialize finished
[ERROR] ME (334540, ffff887bdbe0, python) :2023-10-13-18:55:54.973.902
[mindspore/lite/src/extendrt/c xx_api/model/model.cc:101] Build] Catch exception:
The pointer[mngl is null.
- Framework Unexpected Exception Raised:
This exception is caused by framework's unexpected error. Please create an issue at
https://gitee.com/mindspore/mindspo re/issues to get help.
- C++ Call Stack: (For framework developers)
mindspore/core/ir/func_g raph.cc:352 free_variables_total
Traceback (most recent call last):
   File "lite chat.py", line 206, in modules
      chat(tokenizer file, prefill mindir_path, decode mindi r_path, lite_config,
device_id)
   File "lite chat.py", line 75, in load model
      model0.build from file(model pathO, mslite.ModelType. MINDIR, context,
config file)
   File "/home/miniconda3/lib/python3.7/site-packages/mindspore_lite/model.py, line
95, in warpper
      return func(*args, **kwargs)
   File "/home/miniconda3/lib/python3. 7/site-packages/mindspore_lite/model .py",
line 235, in build_from_file
      raise RuntimeError(- f"build from file - failed! Error is -
(ret.Tostring()}")
RuntimeError: build from file failed! Error is Common error code.
[TRACEI GE(334540,python):2023-10-13-18:55:55.355.775 [status:INIT] [ge
api .cc:3621334540 GEFinalize: GEFinalize start
[TRACEI GE(334540,python):2023-10-13-18:55:55.355.910 [status :RUNNING]
Tge_api.cc:3731334540 GEFinalize:Finalizing environment
[TRACE] GE(334540,python):2023-10-13- 18:55:58.021.147 [status:SToP]
[ge_api .cc:4011334540 GEFinalize:GEFinalize finished
```

3. 根因分析

通过设置日志等级

```
export ASCEND_SLOG_PRINT_TO_STDOUT=1
export ASCEND_GLOBAL_LOG_LEVEL=1
```

确定报错的原因，发现是内存不足。

```
DewMemAllacHgePageManaged:[LOAD][LOAD][drv ap1] halWemAlloc failed:
sLze=1572400(Byte), type=2, modJleId=45, drvFlag=3242591731706905600, drwRetCode=6!
```

4. 解决方案

经过实验发现，对于 mindspore2.2 之前的版本，通过将环境变量 MS_GE_TRAIN 去除，或者将 MS_GE_TRAIN 设置为 0，可以成功加载。

6.9 Mindspore 在自回归推理时的精度对齐设置

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend/GPU/CPU

软件环境: MindSpore 版本: 2.1

执行模式： 不限

2. 报错信息

2.1 问题描述

自回归推理时，不会使用 KV cache，每次生成的 token 会拼接在当前输入的最后位置，作为下一次的输入。在 mindspore 中通过配置 use_past 设为 False 来开启自回归推理，在 huggingface 的 transformer 中模型配置文件 config.json 中配置项 use_cache 表示是否开启 KV cache，但是直接设置 use_cache 为 false 并不会生效。

```
{
    _name_or_path": "bigcode/starcoder"
    "activation_function": "gelu"
    "architectures": [
    "GPTBigCodeForCausalLM'
    "attention softma x in fp32": true,
    "attn pdrop": 0.1,
    "bos token id": 0,
    "embd pdrop": 0.1,
    "eos_token_id": 0,
    "inference_runner": 0,
    "initializer_range": 0.02,
    "layer_nom_epsilon": 1e-05,
    "max_batch_size": null,
    "max_sequence_length": null,
    "model_type": "gpt_bigcode",
    "multi_query": true,
    "n_embd": 6144,
    "n_head": 48,
    "n_inner": 24576,
    "n_layer": 40,
    "n_positions": 8192
    "pad_key_length": true,
    "pre_allocate_kv_cache": false,
    "resid_pdrop": 0.1,
    "scale_attention_so ftmax_in_fp32": false,
    "scale_attn weights": true,
    "summary_ activation": null,
    "summary first  dropout": 0.1,
    "summary_proj_to labels": true,
    "summary_type": "cls index",
    "summary_use_ proj": true,
    "torch dtype": "float16",
    "transfommers_version": "4.30.0.dev0",
    "use cache": true,
    vdLiudLe rumier_.Lnput": true,
```

```
    "vocab size": 49153
}
```

3. 解决方案

需要在 model.generate 方法中也传入 use_cache=False。

```
with torch.no_grad():
    generation output = model -generate(
        input ids=input ids,
        generation config=generati on_config,
        return dict in genera te=True,
        output scores=True,
        max new tokens=max new tokens,
        output hidden states=True,
        use cache=False
    )
```

6.10 MindSpore 大模型打开 pp 并行或者梯度累积之后 loss 不溢出也不收敛

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.0

执行模式： 不限

2. 报错信息

2.1 问题描述

使用 MindSpore 大模型打开 pp 并行或者梯度累积之后 loss 不溢出也不收敛。

3. 根因分析

应该是 loss 计算方式有问题，实际溢出但是显示为 False

4. 解决方案

手动减小 loss_scale，按经验可尝试设置为 65536、2048 等。微调时可同步设置 beta2 为 0.999。

6.11 Ascend 上用 ADGEN 数据集评估时报错 not support in PyNative RunOp!

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.0

执行模式： 不限

2. 报错信息

2.1 问题描述

在 Ascend 上用 ADGEN 数据集评估时报错 not support in PyNative RunOp!

2.2 报错信息

```
[WARNING] GE_ADPT (2042778, ffffaf6cfc20,python) :2023-10-12-17:27:01.018.444
[mindspore/ccsrc/transfo rm/graph_ir/utils.cc:276] CheckAndGetGraphRunner ] Can not
find init_subgraph. kernel_graph_1 sub graph, don't need data init subgraph in
INFER mode.

Traceback (most recent call last):
   File "wizardcoder/run_wi zardcoder.py", line 150, in <module>
      device_id=args.device_id)
   File "wizardcoder/run_wiza rdcoder.py", line 99, in main
      task.evaluate(eval_chec kpoint=config.model .
model_config.checkpoint_name_or path)
   File "/home/wizardcoder/ 1_wizardcoder-mindformers-916/mindformers/trainer/t
rainer.py", line 589, in evaluate
      is full config=True, **kwargs
   File "/home/wizardcoder/1_wizardcoder-mindformers-
916/mindformers/trainer/causal_lan guage_ modeling/causal_language_modeling.py",
line 156, in evaluat
      **kwargs)
   File * /home/wizardcoder/1 wizardcoder-mindformers -
916/mindformers/trainer/base_trainer.py", line 713, in evaluatel_process
      dataset sink mode=con fig.runner_config.sink mode)
   File "/ root/miniconda3/envs/iib/python3.77site-
packages/mindspore/train/model.py", line 1470 in eval
      eval result = self. -eval dataset sink process (valid dataset, list callback,
cb params)
   File "/root/miniconda3/envs/lib/python3.7/site-packages/mindspore/train/model.
py", line 1329, in _eval_dataset_sink_process
      self._update _metrics(outputs)
   File "/root/miniconda3/envs/lib/python3.7/site-
packages/mindspore/train/model.py", line 394, in _update metrics
      metric.update(*outputs)
   File "/home/wizardcoder/1 wizardcoder-mindformers -
916/mindformers/core/metric/metric .py", line 545, in update
      logits = self.reshape( logits[:: :-1, ::], (batch_size * (seq_length - 1), -
1))
   File "/root/miniconda3/envs/lib/python3.7/site-
packages/mindspore/ops/primitive .py", line 314, in__call__
      return _run_op(self, self.name, args)
   File "/root/miniconda3/envs/iib/python3.7/site-
packages/mindspore/ops/primitive .py", line 907, in _run_op
      stub = _pynative_executor . run_op_async (obj, op_name, args)
   File „/root/miniconda3/envs/python3. 7/site-packages/mindspore/common/api .py",
line 1275, in run_op_async
      return self. executor. run_op_async(*args)
RuntimeError: GE backend is not support in PyNative RunOp!
-----------------------------------------------------------
```

```
- C++ Call Stack: (For framework developers)
------------------------------------------------------------
mindspore/ccsrc/runtime/pynat ive/op compiler.cc:203 Chec kBackendPolicy
```

3. 根因分析

PerplexityMetric 有图外算子导致。

4. 解决方案

将 PerplexityMetric 替换为 ADGENMetric。

6.12 qwen1.5_1.8B 推理出现回答混乱问题及解决

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 报错信息

qwen1.5_1.8B 推理出现回答混乱的情况

```
2024-05-09 10:14:49,029 - mindtormers
 [mindtormers/generation/text_generator.py:1103] - INFO - Generation Contig is:
{'max_length': 8192, rax_new_tokens': None, "num beams': 1, . 'do sample': True.
'use past': True, 'temperature': 1.0, 'top k': 1, 'top_p': 0.8,
"repetition_penalty': 1, 'encoder_repetition_penalty':1.0, 'renormalize_logits":
False ,'pad_token_id': 151645,' bos_token_id': 1, 'eos_token_ id': 151645, _from
model config': True}
2024-05-09 10:14:49,030 - mindformers[mindformers/generation/text_generator.py:174]
- INFO - The generation mode will be **SAMPLE**
2024-05-09 10:14:49,031 - mindformers[mindformers/generat ion/text_generator.py:551]
- WARNING - max_length 8192 can not exceeds model seq_length 2048, set max_length =
seq_length.
2024-05-09 10:18:36,522-- mindformers[mindformers/generation/text_generator .py:724]
- INFO - total time: 227.4914104938507 s; generated tokens: 2039 tokens; generate
speed: 8.962975769386757 tokens/s
2024-05-09 10:18:36,685 - mindformers[mindformers/trainer/base_t rainer.py:940] -
INFO - output result is: [{'text_generat ion_text': ['帮助我制定一份去上海的旅游攻略，以
上海的旅游规划，以上海的旅游攻略，以上海的旅游攻略， 以上海的旅游攻略，以上海的旅游景点，以景点景点
景点景  购买，游玩， 门票购买，门票购买门票购实，门票购 实，门票购买，购实门票购实门票购实门票购实，
购买， 购买购实门熏购实购实购买购买门票购实购买购买购实购实购买购实购买购买购买购买购购买购买购买购
买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买门票购买 门栗购买门票
购买门票购买门栗购买门票购买门票购买门票购买门票门票门票门票 门票门票门票门票门票门票 门票门票门票购
买门票购买门票购买门票购买门票购买门票购买门票购买门票购买门票购买  门票购买门票购买门票购买门票购
买门票购买门票购买门票购买门票购买门票购买门  票购买门票购买门票购买门票购买门   票购买门票购买
门票购买 门票购买地购买地点购买门票购买门票购买地点购买地点购买门票购买地点购买门票购买门票购买门票
购买地点购买地点购买门票购买地点购买地点购买门票购买购买购购买地点购买购买地点购买购买购买购买购买购
买购买地点购买购买购买购买购买地点购买购买购买购买购买购买购买购买购买购买购买地点购买地点购买购买购
买购买地点购买购买购买购买购买购买购买购买购买购买购买购买地点购买购买购买购买购买购买购买地点购买购买
买购买地点购买地点购买购买购买购买购买购买购买购买购买地点购买地点购买购买购买购买购买购买购买购买购买
```

```
买购买购买购买购买购买购买购买购买地点购买购买购买购买购买购买购买购买购买购买购买购买购买费购买
购买购买购买购买贝构买购买购买购买购买购买购买费购买购买购买购买购买购买购买费购买购买购买购
买购买购买购买购买购买购买地点购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购
买地点购买购买购买购买购买购买购买购买购买购买购买地点购买购买购买购买购买购买购买购买购买购
买购买地点购买购买购买购买地点购买购买购买购买购买购买购买购买购买购买地点购买购买购买购
买购买购买购买购买购买！购买买购买地点购买购买购买！购买购买地点购买购买购买购买购买地点购买购买购
买购买购买地点购买购买购买购买购买购买购买购买购买购买购买地点购买购买购买购买地点购买购买购
买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购
买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购
买购买购买购买购买购买购买购买费购买购买购买购买购买购买费购买购买购买购买购买费购买购买购买
购买购买购买费购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购
买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购
买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购
买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购
买购买购买购买购买购买购买购买费购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买
购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买
购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买
购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买费买购买购买购买购买购买费购买购
买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购
买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购
买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购买购
买购买购买购买购买购买购买购买购买购买购买购买购买购买购买买
```

3. 解决方案

属于环境问题，装错了 nnae;改为 toolkit 后推理正常，使用 nnae 进行推理时需要先设置：

```
export MS_GE_TRAIN=0
export MS_ENABLE_GE=1
export MS_ENABLE_REF_MODE=1
```

yaml 根据 hg 上的 Transformer 的 confing.json 来修改

6.13 单机 4 卡分布式推理失报错 RuntimeError: Ascend kernel runtime initialization failed. The details refer to 'Ascend Error Message'.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 报错信息

README：
https://gitee.com/mindspore/mindformers/blob/dev/research/deepseek/deepseek.md

单机 4 卡分布式推理失败，出现 out of memory 报错。报错信息如下：

```
[WARNING] DISTRIBUTED(2517382,ffff92cf20b0,python):2024-05-08-15:44:00.654.709
[mindspore/ccsrc/distributed/rpc/tcp/tcp_comm.cc:464] Connect] Waiting for the
state of the connection to 127.0.0.1:8118 to be connected...Retry number: 1
[WARNING] DISTRIBUTED(2517382,ffff92cf20b0,python):2024-05-08-15:44:01.659.911
[mindspore/ccsrc/distributed/cluster/cluster_context.cc:194] BuildCluster] Topology
build timed out., retry(1/200).
[WARNING] DISTRIBUTED(2517382,ffff92cf20b0,python):2024-05-08-15:44:04.660.169
[mindspore/ccsrc/distributed/cluster/cluster_context.cc:196] BuildCluster] Cluster
is successfully initialized.
[WARNING] DISTRIBUTED(2517382,ffff92cf20b0,python):2024-05-08-15:44:04.660.685
[mindspore/ccsrc/distributed/cluster/cluster_context.cc:260] PostProcess] This node
0 rank id: 0
[WARNING] DEVICE(2517382,ffff92cf20b0,python):2024-05-08-15:44:04.850.464
[mindspore/ccsrc/plugin/device/ascend/hal/device/ascend_memory_adapter.cc:95]
Initialize] Reserved memory size for other components(1040187392) is less than
recommend size(4073433600), It may lead to Out Of Memory in HCCL or other
components, Please double check context key
'variable_memory_max_size'/'max_device_memory'
2024-05-08 15:44:44,925 -
mindformers[mindformers/tools/cloud_adapter/cloud_monitor.py:43] - ERROR -
Traceback (most recent call last):
  File "/home/jenkins0/csj/mindformers/mindformers/core/context/build_context.py",
line 120, in init_context
    init()
  File "/home/miniconda3/envs/ci/lib/python3.7/site-
packages/mindspore/communication/management.py", line 188, in init
    init_hccl()
RuntimeError: Ascend kernel runtime initialization failed. The details refer to
'Ascend Error Message'.

----------------------------------------------------
- Framework Error Message:
----------------------------------------------------
Malloc device memory failed, size[64424509440], ret[207001], Device 0 Available HBM
size:65464696832 free size:65165885440 may be other processes occupying this card,
check as: ps -ef|grep python

----------------------------------------------------
- Ascend Error Message:
----------------------------------------------------
EL0004: 2024-05-08-15:44:08.292.766 Failed to allocate memory.
       Possible Cause: Available memory is insufficient.
       Solution: Close applications not in use.
       TraceBack (most recent call last):
       rtMalloc execute failed, reason=[driver error:out of
memory][FUNC:FuncErrorReason][FILE:error_message_manage.cc][LINE:53]
       alloc device memory failed, runtime result =
207001[FUNC:ReportCallError][FILE:log_inner.cpp][LINE:161]
```

3. 解决方案

实际使用的内存大小超出了设备预先设置的可用内存大小，可以修改 yaml 参数 max_device_memory，提升设备预设可用内存。

6.14 使用单卡 Ascend910 进行 LLaMA2-7B 推理,速度缓慢

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： Graph

2. 报错信息

使用单卡进行 LLaMA2-7B 推理，推理速度只有 3.75token 每秒。

设备是 910，MindSpore==2.2.11， mindformers==1.0.0

3. 解决方案

开启增量推理，开启后推理速度正常

```
IMF0:  10.254.0.1:13452 - "POST /generate HTTP/2.1" 200  0%
2024-04-23 03:55:32,127 - mindformers[mindformers/generation/text_generator.py:478]
- INFO - total time: 21.01030683517456 s; generated tokens: 504 tokens; generate
speed: 23.988226538235256 tokens/s
2024-04-23 03:55:32,142 - mindformers[mindformers-1-0.0/chat_web/predict
process.py:122] - INFO - 0 card generate output: it ts a city that has everything.
It is a city of contrasts, where you can find the most  modern skyscrapers and the
sost ancient teaples in the same street.
```

6.15 MindSpore 大模型微调时报溢出及解决

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 报错信息

使用模型在 Ascend 上微调时持续报溢出。

```
dspore/train/model.py:653] In dataset_sink mode (dataset_size % sink_size) should
equal to 0, it is suggested to pad/drop data or adjust size
 - INFO - { Epoch:[ 1/ 2], step:[ 2/ 3915], loss: 1.653, per_step_time: 64622ms, Ir:
0.0, overflow cond: True, loss_scale: 16384.0
 - INFO -    0.1%
 - INFO - { Epoch:[ 1/ 2], step:[ 4/ 3915], loss: 1.510, per_step_time: 741ms, Ir:
0.0, overflow cond: True, loss_scale: 4096.0
 - INFO -    0.1%
 - INFO - { Epoch:[ 1/ 2], step:[ 6/ 3915], loss: 1.606, per_step_time: 753ms, Ir:
0.0, overflow cond: True, loss_scale: 1024.0
 - INFO -    0.1%
 - INFO - { Epoch:[ 1/ 2], step:[ 8/ 3915], loss: 1.309, per_step_time: 741ms, Ir:
0.0, overflow cond: True, loss_scale: 256.0
```

```
 - INFO -     0.1%
 - INFO - { Epoch:[ 1/ 2], step:[10/ 3915], loss: 1.233, per_step_time: 750ms, Ir:
0.0, overflow cond: True, loss_scale: 64.0
 - INFO -     0.2%
 - INFO - { Epoch:[ 1/ 2], step:[12/ 3915], loss: 2.089, per_step_time: 739ms, Ir:
0.0, overflow cond: True, loss_scale: 16.0
 - INFO -     0.2%
 - INFO - { Epoch:[ 1/ 2], step:[14/ 3915], loss: 1.446, per_step_time: 743ms, Ir:
0.0, overflow cond: True, loss_scale: 4.0
 - INFO -     0.2%
 - INFO - { Epoch:[ 1/ 2], step:[16/ 3915], loss: 1.639, per_step_time: 741ms, Ir:
0.0, overflow cond: True, loss_scale: 1.0
 - INFO -     0.2%
```

3. 解决方案

```
#设置是否校验中间过程溢出
export MS_ASCEND_CHECK_OVERFLOW_MODE=INFNAN_MODE
```

在遇到持续溢出问题时可以尝试。

INFNAN 模式会忽略过程溢出，只要结果是非溢出的 就会继续训练。

饱和模式是有中间过程溢出，就会上报溢出，算法就不更新 loss。

6.16 MindSpore 大模型在线推理速度慢及解决方案

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： Graph

2. 报错信息

模型在线推理速度很慢，只有 0.x tokens/s，可能有哪些原因？

3. 解决方案

仓上的模型在线推理都经过了性能验证，增量推理的性能基本都有 10+ tokens/s 的性能；出现推理异常慢的现象，通常是因为使用上的配置问题，没有正确获得推理性能；

1.推理需使用图模式，mindspore.set_context(mode=0)；MS2.x 的版本，默认执行模式是 pynative 模式，性能较差，需手动设置图模式运行。

2.确认模型开启了增量推理，模型配置项中，将 use_past 配置项设为 True 以开启增量推理，获得更好的性能；自回归推理存在大量冗余计算，在长 seq 的场景下推理性能较差。

3.多次运行推理，排除首次推理的性能数据；在线推理需要对模型进行编图，加载模型后首次进行推理时性能会受图编译影响，后续重复调用推理，可复用首次编译的图，得到正确的推理性能。

6.17 Llama 推理报参数校验错误 TypeError: The input value must be int. but got 'NoneType.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： Graph

2. 报错信息

Llama 推理报参数校验错误。

```
(MindSpore) [ma-user transfer_checkpoint]$python test_baichuan_78.py
Traceback (most recent call last):
   File "/home/ma-user/work/mindformers/transfer_checkpoint/test_baichuan_78.py",
line: 16, in <module>
       baichuan_model - LlamaForCausalLM(
   File "/home/ma-user/anaconda3/envs/MindSpore/1ib/python3.9/site-
packages/mindformers/models/1lama/1lama .py", line 253, in __init__
       self.model - LlamaModel (config=config)
   File "/home/ma-user/anaconda3/envs/MindSpore/1ib/python3.9/site-
packages/mindformers/models/1lama/1lama.py", line 94, in __init__
       Validator.check_positive_int(config.batch_size)
   File "/home/ma-user/anaconda3/envs/MindSpore/1ib/python3.9/site-
packages/mindspore/_checkparam py", line 331, in check_positive_int
       return _check number(arg value, 0, GT,.int, arg-name, prim name)
   File -/home/ma-user/anaconda3/envs/Mindspore/1ib/python3.9/site-
packages/mindspore/ checkparam.py", line 220, in check number
       check param()
   File -/home/ma-user/anaconda3/ervs/MindSpore/1ib/python3.9/site-
packages/mindspore/ checkparam.py",line 208, in check param
       raise TypeError( f"{prim info} must be farg type. __name__}, but got
{type(arg_value).__name}'")
TypeError: The input value must be int. but got 'NoneType.
```

3. 解决方案

出错位置在做参数校验，上面是对 config.batch_size 做校验，检验结果为 config.batch_size 为 int，但是实际是 NoneType，表示传入的值是一个空值；需要检查 config.batch_size 是否配置，在 yaml 中配置位置是否错误，值是否正确；其他参数校验错误，也可参照以上解决办法。

6.18 MindSpore 模型报错 Reason: Memory resources are exhausted.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 报错信息

```
File "/home/ma-user/modelarts/user-job-dir/train.py", line 309, in
run_train_no_pipeline
    model.train(actual_epoch_num, ds, callbacks=callback,
sink_size=args_opt.sink_size, dataset_sink_mode=True)
  File "/home/ma-user/anaconda/lib/python3.7/site-
packages/mindspore/train/model.py", line 1065, in train
    initial_epoch=initial_epoch)
  File "/home/ma-user/anaconda/lib/python3.7/site-
packages/mindspore/train/model.py", line 113, in wrapper
    func(self, *args, **kwargs)
  File "/home/ma-user/anaconda/lib/python3.7/site-
packages/mindspore/train/model.py", line 619, in _train
    cb_params, sink_size, initial_epoch, valid_infos)
  File "/home/ma-user/anaconda/lib/python3.7/site-
packages/mindspore/train/model.py", line 702, in _train_dataset_sink_process
    outputs = train_network(*inputs)
  File "/home/ma-user/anaconda/lib/python3.7/site-packages/mindspore/nn/cell.py",
line 641, in __call__
    out = self.compile_and_run(*args, **kwargs)
  File "/home/ma-user/anaconda/lib/python3.7/site-packages/mindspore/nn/cell.py",
line 964, in compile_and_run
    return _cell_graph_executor(self, *new_args, phase=self.phase)
  File  "/home/ma-user/anaconda/lib/python3.7/site-packages/mindspore/common/api.py",
line 1548, in __call__
    return self.run(obj, *args, phase=phase)
  File  "/home/ma-user/anaconda/lib/python3.7/site-packages/mindspore/common/api.py",
line 1587, in run
    return self._exec_pip(obj, *args, phase=phase_real)
  File  "/home/ma-user/anaconda/lib/python3.7/site-packages/mindspore/common/api.py",
line 110, in wrapper
    results = fn(*arg, **kwargs)
  File  "/home/ma-user/anaconda/lib/python3.7/site-packages/mindspore/common/api.py",
line 1567, in _exec_pip
    return self._graph_executor(args, phase)
RuntimeError: Exec graph failed
----------------------------------------------------
- Ascend Error Message:
----------------------------------------------------
EI0007: Failed to allocate resource[qp] with info
[rdmaHandle:187651097805232flag:0qpMode:3281438645626752]. Reason: Memory resources
are exhausted.
       Possible Cause: Failed to allocate memory or the Notify register due to
resource insufficiency.
       TraceBack (most recent call last):
       Call ops_kernel_info_store LoadTask
fail[FUNC:Distribute][FILE:hccl_task_info.cc][LINE:320]
       Call ops_kernel_info_store unloadTask
fail[FUNC:Release][FILE:hccl_task_info.cc][LINE:657]
(Please search "Ascend Error Message" at https://www.mindspore.cn for error code
description)
```

```
----------------------------------------------------
- C++ Call Stack: (For framework developers)
----------------------------------------------------
mindspore/ccsrc/plugin/device/ascend/hal/hardware/ge_graph_executor.cc:997 RunGraph
```

3. 根因分析

内存分配失败，这里指的是 hccl 所需内存不够，导致失败。

4. 解决方案

这里 max_device_memory 设置过大导致系统分配给 hccl 的内存不够，尝试减小 max_device_memory 参数的值。

```
context.set_context(max_device_memory=max_device_memory)
```

6.19 MindSpore2.2.10 ge 图模式报错: Current execute mode is KernelByKernel, the processes must be launched with OpenMPI or ...

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend910

软件环境: MindSpore 版本: 2.2.10

执行模式： 静态图

2. 报错信息

MindSpore2.2.10 ge 图模式误跑到了 KBK 流程。

```
RuntimeError: Current execute mode is KernelByKernel, the processes must be
launched with OpenMPI or ...
```

3. 根因分析

开启 INFO 日志，定位到走到 KBK 的原因为存在 dynamic scalar ops，通过 IR 图定位到业务代码中有使用 int()的强制转换，调用了 scalar_cast 算子所导致。

```
[INFO] PIPELINE(2963556, ffff8c4e8010,Python):2024-04-08-09:52:08.630.899
[mindspore/ccsrc/pipeline/ji t/ps/action.cc:1258] SetRunModel Run graph mode with
kernel by kernel cause graph exist dynamic scalar ops.
[CRITICAL] PIPELINE(29635s6,ttttöc4e8010, python) : 2024-04-08-09:52:08.633.509
[mindspore/ccsrc/pipeline/jit/ps/action.cc:1328] SetRunModel Current execute mode
is kernel bynel, the processes must be launched with OpenMPI or Dynamic
Cluster(Without setting MS_HCCL_CM_INIT to 1)
W29999: [Init][LoadAscendCustomOppPath]
ASCEND_CUSTOM_OPP_PATH[/cache/yzs/bev_ms/vendors/ms] is not exist or not abstract
path. [FUNC: AddCustomOpstoreContent][FILE:configuion.cc][LINE:641]

This exception is caused by framework's unexpected error. Please create an issue at
https://gitee. com/mindspore/mindspore/issues to get help.
[INFO] DEBUG(2963556, ffff8c4e8010,python):2024-04-08-09: 52:08.633.582
```

```
[mindspore/ccsrc/pipeline/jit/ps/debug/trace.cc:117] TraceGraphEval] Length of
analysis graph stack empty.
[INFO] DEBUG(2963556, ffff8c4e8010, python) :2024-04-08-09:52:08.633.641
[mindspore/ccsrc/pipeline/j it/ps/debug/trace.cc:418] GetEvalStackInfo] Get graph
analysis informatio egin
[INFO] DEBUG(2963556, ffff8c4e8010,python) :2024-04-08-09: 52:08.633.670
[mindspore/ccsrc/pipeline/jit/ps/debug/trace .cc:421] GetEvalStackInfo] Length of
analysis informatio tack is empty.
[INFO] PIPELINE(2963556,ffff8c4e8010,python):2024-04-08-09: 52:08.634.289
[mindspore/cc src/include/common/utils/python_utils . h:368] HandleExceptionRethrow]
Caught exceptic Current execute mode is kernelbykernel, the processes must be
launched with OpenMPI or Dynamic Cluster(Without setting MS_HCCL_CM_INIT to 1)

  Ascend Warning Message:
W29999: [Init][LoadAscendCustomOppPath]
ASCEND_CUSTOM_OPP_PATH[/cache/yzs/bev_ms/vendors/ms] is not exist or not abstract
path.[FUNC:AddCustomOpStoreContent][FILE: configuion.cc][LINE:641]

- Framework Unexpected Exception Raised:
This exception is caused by framework's unexpected error. Please create an issue at
https://gitee. com/mindspore/mindspore/issues to get help.

- C++ Call Stack: (For framework developers)
mindspore/ccsrc/pipeline/jit/ps/action.cc:1328 SetRunMode
c = (int(r[0][0] / x*4, r[0][0]*fov / 1))
```

4. 解决方案

分析代码中 int 类型转换为冗余操作，去掉后无报错。

```
c = (r[0][0] / x*4, r[0][0]*fov / 1)
```

6.20 baichuan2-13b 算子溢出 loss 跑飞问题和定位

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.11

执行模式： 不限

2. 报错信息

2.1 问题描述

baichaun2-13b 多机开 pipeline 后 loss 先下降，后续持续溢出，loss 跑飞。

3. 根因分析

dump 溢出算子

溢出检测设置：https://www.mindspore.cn/tutorials/experts/zh-CN/r2.2/debug/dump.html#%E5%BC%82%E6%AD%A5dump 设置只 dump 溢出算子。

op_debug_mode：该属性用于算子溢出调试，设置成 0，表示不开启溢出；设置成 1，表示开启 AiCore 溢出检测；设置成 2，表示开启 Atomic 溢出检测；设置成 3，表示开启全部溢出检测功能。dump 成功后分析第一个溢出的算子：

```
SqrtGrad.Gradients_Default_network-MFTrainOneStepCell_network-_VirtualDatasetCell_
backbone-Baichuan13BV2ForCausalLM_Im head-NormHead..
SqrtGrad.Gradients_Default_network-MFTrainOneStepCell_network-_VirtualDatasetCell_
backbone-Baichuan13BV2ForCausalLM_Im head-NormHead..
SqrtGrad.Gradients_Default_network-MFTrainOneStepCell_network-_VirtualDatasetCell_
backbone-Baichuan13BV2ForCausalLM_Im head-NormHead..
SqrtGrad.Gradients_Default_network-MFTrainOneStepCell_network-_VirtualDatasetCell_
backbone-Baichuan13BV2ForCausalLM_Im head-NormHead..
SqrtGrad.Gradients_Default_network-MFTrainOneStepCell_network-_VirtualDatasetCell_
backbone-Baichuan13BV2ForCausalLM_Im head-NormHead..
SqrtGrad.Gradients_Default_network-MFTrainOneStepCell_network-_VirtualDatasetCell_
backbone-Baichuan13BV2ForCausalLM_Im head-NormHead..
Opdebug.Node_OpDebug.221.19.1705388945730580.output.0.json
Opdebug.Node_OpDebug.221.19.1705388725219277.output.0.json
```

通过 dump 溢出算子，识别到是 sqrt 反向溢出

单算子复现溢出场景：

```
@Plugins.golden.register([ "sqrt_grad"])
def _sqrt_grad(conte xt: UniversalTe stcasestructure):
    inputx = context.input_arrays[ø]
    inputy = context.input_arrays[1]
    div_val = np.divide( inputy, inputx)
    res = div_val * 0.5
    return [res]
>>> input1
array([[ 6.265e+02],
      [-2.404e+00],
      [-1.394e+00],
      ...,
      [-5.176e-01],
      [-3.635e+00],
      [-1.166e+00]], dtype=float16)

>>> input1.max()
47580.0
```

算子溢出导致 loss 跑飞。

4. 解决方案

针对溢出算子 cast 成 float32 或者 bfloat16，转 flaot32 性能可能会劣化，推荐转 bfloat16。

6.21 MindSpore 训练异常中止：Try to send request before Open()、Try to get response before Open()、Response is empty

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.0&2.2.10

执行模式： Graph

2. 报错信息

MindSpore 训练异常中止：Try to send request before Open()、Try to get response before Open()、Response is empty。

```
Sampling with EulerEDMSampler for 50 steps: 0%| | 0/50 [00:00<?, ?it/s]
Sampling with EulerEDMSampler for 50 steps: 0%| | 0/50 [00:00<?, ?it/s]
Traceback (most recent call last):
   File "/home/ma-user/modelarts/user-job-
dir/stable_diffusion_xl/demo/pangu3_infer_server_modelarts.py", line291, in <module>
       sample(args)
   File "/home/ma-user/modelarts/user-job-
dir/stable_diffusion_xl/demo/pangu3_infer_server_modelarts.py", line274, in sample
       version dict,
   File "/home/ma-user/modelarts/user-job-dir/stable diffusion_xl/demo/pangu3
infer_server_modelarts.py", line207, in run_txt2img
       samples z = sampler(model, high timestamp model, randn, cond=c, uc=uc, other
c=other c)
   File "/home/ma-user/modelarts/user-job-
dir/stable_diffusion_xl/gm/modules/diffusionmodules/sampler.py", line 218, in
_call_
       gamma,
   File "/home/ma-user/modelarts/user-job-
dir/stable_diffusion_xl/gm/modules/diffusionmodules/sampler.py", line189, in
sampler_step
       denoised = self.denoise(x,model, sigma hat, cond, uc, other c)
   File "/home/ma-user/modelarts/user-job-
dir/stable_diffusion_xl/gm/modules/diffusionmodules/sampler.py", line 54, in
denoise
       c skip, c_out, c_in, c noise = model.denoiser(sigmas, noised_input.ndim)
   File "/home/ma-user/anaconda/lib/python3.7/site-packages/mindspore/nn/cell.py",
line 664, in _call_
       raise err
   File "/home/ma-user/anaconda/lib/python3.7/site-packages/mindspore/nn/cell.py",
line 661, in _call_
       pynative executor.end graph(seIf, output, *args, **kwargs)
   File "/home/ma-user/anaconda/lib/python3.7/site-
packages/mindspore/common/api.py", line 1304, in end_graph
       self. executor.end graph(obj, output, *args, *(kwargs.values0))
       RuntimeError: Try to send reguest before Open().For more details
```

3. 根因分析

此类问题的直接原因一般是算子编译的子进程挂了或者调用阻塞卡住导致的超时，可以从以下几个方面进行排查：

1. 检查日志，在这个错误前是否有其他错误日志，如果有请先解决前面的错误，一些算子相关的问题（比如昇腾上 TBE 包没装好，GPU 上没有 nvcc）会导致后续的此类报错；
2. 如果有使用图算融合特性，有可能是图算的 AKG 算子编译卡死超时导致，可以尝试关闭图算特性；
3. 在昇腾上可以尝试减少算子并行编译的进程数，可以通过环境变量 MS_BUILD_PROCESS_NUM 设置，取值范围为 1~24(建议修改为 16， 正常 Ascend 机器 host cpu192 个线程，云上做了限制，实际没 192 个。框架默认 24 个，因此报错)；
4. 检查主机的内存和 cpu 占用情况，有可能是主机的内存和 cpu 占用过高，导致算子编译进程无法启动，出现了编译失败，可以尝试找出占用内存和 cpu 过高的进程，对其进行优化；
5. 如果是在云上的训练环境遇到这个问题，可以尝试重启内核。

6.22 Ascend 910 训练脚本刚运行就报错：RuntimeError: Initialize GE failed!

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.0

执行模式： 不限

2. 报错信息

训练脚本刚运行就报错：RuntimeError: Initialize GE failed!

```
   File "./train_test.py", line 136, in run
      mtl_fwk « MultiTaskFramework(config-config)
   File "/train-worker1 -log/g@@567577/bev/uvpsandbox-
api/uvpsandbox/uvpsandbox/framework.py", line 122, in init
      self._build customized(self.cfg.task)
   File "/train-worker1 -log/g@0567577/bev/uvpsandbox-
api/uvpsandbox/uvpsandbox/framework.py", line 352, in build customized
      train_data_utils = build_dataloader(task_cfg.dataset.train.nme,
task_cfg.dataset.train, Inue) \
   File "/train-worker1 -log/g00567577/bev/uvpsandbox-
api/uvpsandbox/uvpsandbox/subnet_context.py", Line 423, in build dataloader
      build_loader_sampler(dataset-builders.dataset.get_builder(name,
dataset_fields)(cfg), is_training-training))
   File "/train-worker1 -log/g00567577/bev/uvpsandbox-
api/uvpsandbox/uvpsandbox/subnet_context.py", line 221, in build loader sampler
      bird_view_feature_ batch « BirdViewFeatureBatchap(dataset .config)
   File "/train-worker1 -log/g00567577 /bev/uvpsandbox-
api/uvpsandbox/uvpsandbox/subnet_context.py", line 137, in init _
      self.lidar_woxel_layer = BindViefeatureBatch(cfg)
```

```
    File "/train-worker1 -
log/g00567577/bev/tasks/bev_task/uvp_module/models/lidar_backbone/bird_view_feature
_batch.py", line 29, in __init_
        self.op_init = BirdViewInit(cfg)
    File "/train-worker! -
log/g@0567577/bev/tasks/bev_task/uvp_module/models/lidar_backbone/lidar_det_ext.py",
line 10, in __init_
        super(BirdViewInit, self).__init__()
    File "/home/ma-user/miniconda3/envs/ms: .@MixPre/lib/python3.8/site-
packages/mindspore/nn/cell.py", line 135, in init _
        init_pipelline()
Runtimetrror: Initialize GE failed!
```

3. 根因分析

是典型的 GE 初始化失败，怀疑磁盘空间不足，查看 df -h， 发现 overlay 的空间占用 100%，因为之前跑 8 卡发生 core dump，其 dump 文件保存在 /tmp/core-file 路径，导致某空间被占满，初始化 GE 报错。删除 core 文件后解决。

6.23 昇腾 910 上算子溢出问题分析

1. 根因分析

由于昇腾上部分算子不支持 fp32（如 Conv，Pooling），或部分算子虽然支持 fp32 但性能极差，不得不切换成 fp16（如 MatMul），导致其必须走 fp16，这增大了训练过程中算子溢出的风险，影响网络最终收敛的精度，这里给出算子溢出分析方法。

- 步骤 1：开启溢出检测
  - 找到 异步 dump 指导（也可以直接访问 https://www.mindspore.cn/tutorials/experts/zh-CN/master/debug/dump.html?highlight=dump#å¼ æ-¥dump）
- 对于溢出检测，需要注意以下配置：

dump_mode：设置为 0，检测到所有算子

path：推荐设置到/home/ma-user/modelarts/outputs/output_url_0/ 目录下，DI 训练时该目录会自动回传数据到 obs，方便后续进行溢出检测分析

iteration：设置为 all，检测所有迭代

op_debug_mode：设置为 3，开启全部溢出检测

- 一个典型的溢出检测 data_dump.json 配置为：

```
{
"common_dump_settings": {
"dump_mode": 0,
"path": "/home/ma-user/modelarts/outputs/output_url_0/bev/dump/",
"net_name": "bevnet",
"iteration": "all",
"saved_data": "tensor",
"input_output": 0,
"kernels": [],
"support_device": [0,1,2,3,4,5,6,7],
```

```
"op_debug_mode": 3,
"file_format": "npy"
}
```

- 步骤 2：查看溢出算子
- 在日志中搜索关键字 overflow infos，如果有算子溢出，则日志中会提示算子所在的代码行：

```
5290 [WARNING] DEVICEC517, ffff1c1b9160,python):2022-12-27-23:50:13.843.123
[mindspore/ccsrc/plugin/device/ascend/hal/device/ascend_kernel_runtime.
cc:683]TaskFailCallback] Node run task overflow, node name:
Gradients/Default/network-MultiTaskTrainonestepCelL/network-
Modulewrapper/detection3d _head-DetHead/det_head-CenterPointHead/head-
CenterHead/task_heads-CellList/0-SeparateHead/reg-SequentialCel1/1-_OutputTo32/_op-
Conv2d/gradconv2D/Cast-op41878Task overflow infos task_id: 306, streamid: 17,
tid:517, device_id: 1, retcode:207003 (aicore over flow
5291 The function call stack:
5292 Corresponding code candidate:
5293  In file /home/ma-
user/mindspore/build/package/mindspore/ops/_grad/gradnn_ops.py:106/    dx =
input_grad(dout, w, x_shape)/
5294 In file /home/ma-user/mindspore/build/package/mi ndspore/ops/_grad/grad
_nn_ops.py: 107/  dw = filter_grad(dout, x, w_shape)/
5295 Corresponding forward node candidate:
5296 - In file /hone/na-user/mindspore/build/package/mi ndspore/nn/layer/ conv.py:
309/output = self.conv2d(x, self.weight)/
5297 In file /home/na-user/mindspore/bui ld/package/mindspore/train/arp.py:671
    return F.cast(self._opCx), mstype. float32)/
5298 In file /hone/na-user/mindspore/build/package/mi
ndspore/nn/layer/container.py:278/ for cell in self.cell_list:/
5299 In file /home/ma-user/rodelarts/user-job-
dir/code/tasks/bev_task/uvp_rodule/models/det_
head/centerpoint/centerpoint_head.py:145/   return {'reg':self.reg(x),
"height':self.height(x), 'dim':self.dim(x),'rot':self.rot(x),
'vel':self.vel(x),"heatrap':self.heatmap(x)}/
5300 In file /home/ra-user/nodelarts/user-job-dir/code/tasks/bev-
task/uvp_module/nodels/det_head/centerpoint/centerpoint_head.py:239/   for task in
self.task_heads:/
5301 In file /home/ma-user/nodelarts/user-job-
dir/code/tasks/bev_task/uvp_module/models/det_head/centerpoint/centerpoint_
head.py:255/ return multi_apply(self.forward_single,feats)/
5302 In file /hone/ma-user/nodelarts/user- job-
dir/code/tasks/bev_task/uvp_module/models/det_head/centerpoint/centerpoint_head.py:
255/return multi_apply(self.forward_single,feats)/
5303 In file /home/ma-user/nodelarts/user-job-
dir/code/tasks/bev_task/uvp_module/models/det_head/centerpoint/centerpoint_head.py:
413/pred_dict = self.head(bev_feat_map)/
5304 In file /home/ma-user/modlelarts/user-job-
dir/code/tasks/bev_task/uvp_module/models/det_head/det_head.py:71/
    det_output,ret_dict = self.det_headCdata_dict)/
5305 In file /hone/na-user/nodelarts/user-job-dir/code/uvpsandbox-
api/uvpsandbox/uvpsandbox/nodule_wrapper. py: 40/    for net, node, inners, node-
tasks in zip(self.nets, self.nodes, self.inner_nodes, self.nodes tasks):/
5306 In file /hone/ma-user/modelarts/user- job-dir/ code/uvpsandbox-api
/uvpsandbox/uvpsandbox/frarework. py: 583/  net_outputs =
self.network(img_data,f"labels":[labels], "img_info":img_info})/  5307 In File
```

```
/hone/ma-user/mindspore/bui ld/package/mindspore/nn/wrap/loss_scale.py:336/     loss
= self.network(*inputs)/
```

- 溢出的算子为 Cast-op41878 该算子为 Cast 算子，scope 名为 Gradients/Default/network-MultiTaskTrainOneStepCell/network-ModuleWrapper/detection3d_head-DetHead/det_head-CenterPointHead/head-CenterHead/task_heads-CellList/0-SeparateHead/reg-SequentialCell/1-_OutputTo32/_op-Conv2d/gradConv2D/Cast-op41878（如果 scope 的名字以 Gradients 开头，说明该算子是反向产生的算子）。根据 The function call stack: 提示，找到该算子对应的代码（反向算子需要先找到对应的正向算子）。
- 步骤 3：分析溢出产生的原因
- 在步骤 2 中，我们可以找到具体溢出的算子和所在代码，这一步我们需要分析溢出产生的原因。首先需要找出溢出算子所在的 rank，下载该 rank 下的 dump 数据，一个标准的 dump 数据结构如下所示：

```
rank_xx
|-- bevnet          # 根目录，根据 data_dump.json 配置生成
|   |-- 0           # 每个 step 产生的 dump 文件，序号代表 step 数
|   |-- 1
|   |-- 17
|   |-- 2
|   `-- constants     # 网络中的常量，数据不随 step 数改变
|-- execution_order   # 网络中算子的执行顺序（如果有多个图，则产生多个文件）
|   |-- ms_execution_order_graph_0.csv
|   |-- ms_execution_order_graph_1.csv
|   |-- ms_execution_order_graph_17.csv
|   |-- ms_execution_order_graph_18.csv
|   |-- ms_execution_order_graph_19.csv
|   |-- ms_execution_order_graph_2.csv
|   `-- ms_global_execution_order_graph_17.csv
`-- graphs          # 网络中算子的图结构（如果有多个图，则产生多个文件）
    |-- ms_output_trace_code_graph_0.ir
    |-- ms_output_trace_code_graph_0.pb
    |-- ms_output_trace_code_graph_1.ir
    |-- ms_output_trace_code_graph_1.pb
    |-- ms_output_trace_code_graph_17.ir
    |-- ms_output_trace_code_graph_17.pb
    |-- ms_output_trace_code_graph_18.ir
    |-- ms_output_trace_code_graph_18.pb
    |-- ms_output_trace_code_graph_19.ir
    |-- ms_output_trace_code_graph_19.pb
    |-- ms_output_trace_code_graph_2.ir
    `-- ms_output_trace_code_graph_2.pb
8 directories, 21 files
```

- 一般我们只关注 rank_xx/bevnet/x/ 下的数据，假设该目录下数据如下所示：

```
-rw-r--r--   1 root root  630912 Dec 28 10:00 Cast. Cast-
op41877.309.17.1672156213076552. input.0.NCHW. npy
-rw-r--r--   1 root root  315520 Dec 28 10:00 Cast. Cast-
op41877.309.17.1672156213076552.output.0.NCHW.npy
-rw-r--r--   1 root root  630912 Dec 28 10:00 Cast.Cast-
op41878.306.17.1672156213045891. input.0.NCHW.npy
-rw-r--r--   1 root root  315520 Dec 28 10:00 Cast.Cast-
op41878.306.17.1672156213045891.output.0.NCHW. npy
-rw-r--r--   1 root root         1577088 Dec 28 10:00 Cast. Cast-
```

```
op41879.63.18.1672156213413591. input.0.NCHW.npy
-rw-r--r--   1 root root  788608 Dec 28 10:00 Cast.Cast-
op41879.63.18.1672156213413591.output.0.NCHW.npy
-rw-r--r--   1 root root             2432 Dec 28 10:00 Conv2DBackpropInput.
Conv2DBackpropInput-op26836.331.17.1672156213185043.input.0.NCHW.npy
-rw-r--r--   1 root root          315520 Dec 28 10:00 Conv2DBackpropInput.
Conv2DBackpropInput-op26836.331.17.1672156213185043.input.1.NCHW.npy
-rw-r--r--   1 root root       10092672 Dec 28 10:00
Conv2DBackpropInput.Conv20BackoropInput-op26836.331.17.1672156213185043.output-
0.NCHW.nDy
```

- 可以看到，Cast-op41878 的溢出数据被 dump 了出来，读取其中的数据，可以发现该算子的输入 104856.29 在 cast 之后变成了 65500.0 ，算子发生了溢出。

```
Python 3.8.6 (default,0ct 6 2020, 03:22:36)
[GCC 7.5.0] on linux
Type "help", "copyright" , "credits" or "1icense" for more information.
>>> inport nunpy as np
>>> np.1oad("cast.Cast-op41878.306.17.1672156213045891. input.0.NCHW.npy").maxO
104856.29
>>> np.load("cast.Cast-op41878.306.17.1672156213045891.output.0.NCHW. npy").maxO)
65500.0
```

- 由于初始时 loss scale 的值比较大，导致初始梯度比较大，这里的溢出也是正常的，可以尝试调小初始 loss scale 解决。

2. 解决方案

1. 对于发生在训练开始，在后期迭代不再出现的溢出，可以忽略或者调小初始 loss scale。这是因为前期梯度较大，训练不稳定，如果配置了 LossScaleManager，这些溢出的 step 会被自动忽略，一般对最终的收敛影响不大。
2. 如果溢出的算子输入数据正常，正常计算时不应该引发溢出，此时请保留算子输入输出数据，需要与具体的算子开发负责人确认，该算子可能存在溢出误报的问题。
3. 对于 MatMul、BatchMatMul、ReduceSum、ReduceMean 等容易引发溢出的算子，可以尝试在算子输入之前对数据进行缩放（缩小 x 倍），在算子计算后的某个合适位置进行复原（放大回 x 倍），这样在保证整个计算数学等价的情况下，避免溢出。

6.24 使用 ops.nonzero 算子报错 TypeError

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend/GPU/CPU

软件环境: MindSpore 版本: 1.9.0

执行模式： 不限

Python 版本: 3.9

操作系统平台: win10

2. 报错信息

2.1 问题描述

```
It raised TypeError: Type Join Failed: dtype1 = Float32, dtype2 = Int64.
```

2.2 报错信息

```
Traceback (most recent call last):
    File "D:lailrw.py", line 35,in <module>
        main()
    File "D:lailrw.py", line 30, in main
        (loss, logits, extra_output),inputs_gradient = grad_fn(x, y, labels)
    File "D: \python3.9\liblsite-packages\mindspore\commonlapi.py",line 594, in
staging_specialize
        out = _MindsporeFunctionExecutor(func, hash_obj,
input_signature,process_obj, jit_config)(*args)
    File "D: \python3.9\liblsite-packages\mindsporelcommonlapi.py", line 98, in
wrapper
        results = fn(*arg, **kwargs)
    File "D:\python3.9\liblsite-packagles\mindsporelcommonlapi.py",line 405, in -
_call_
        phase = self.compile(args_list, self.fn.-_name-
    File "D: \python3.9\liblsite-packages\mindspore\commonlapi.py",line 379, in
compile
        is_compile = self._graph_executor.compile(self.fn, compile_args, phase,
True)
TypeError: Type Join Failed: dtype1 = Float32, dtype2 = Intó4.
For more details, please refer to
https://www.mindspore.cn/search?inputValue=Type%20Join%20Failed

Inner Message:
This: AbstractScalar(Type: FLoat32, VaLue: AnyValue, Shape: Noshape),
other:AbstractScalar(Type: Intó4, Value: AnyValue, Shape: NoShape). Please check
the node
The function call stack:
In file D:\python3.9\liblsite-packages\mindsporelops\compositelbase.py:527 return
grad_(fn, weights)(*args)/

----------------------------------------------------------
- C++ Call Stack: (For frameworkdevelopers)
----------------------------------------------------------
mindspore\corelabstractlabstract_value.cc:65 TypeJoinLogging
```

2.3 脚本信息

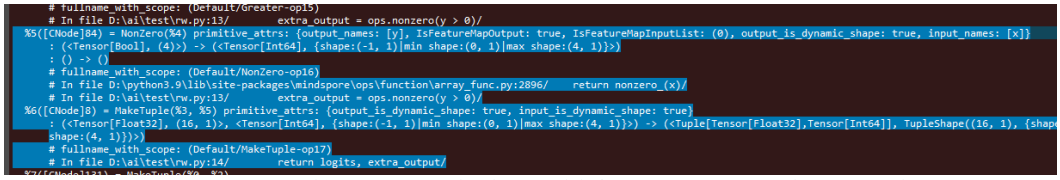
```
import numpy as np
import mindspore as ms
import mindspore.ops as ops
from mindspore import Tensor, nn, context

class Graph(nn.Cell):
    def __init__(self):
        super().__init__()
        self.net = nn.Dense(10, 1)
    def construct(self, x, y):
        logits = self.net(x)
        extra_output = ops.nonzero(y > 0)
        return logits, extra_output
```

```
def main():
    context.set_context(mode=context.GRAPH_MODE)
    net = Graph()
    x = Tensor(np.random.randn(16, 10), dtype=ms.float32)
    y = Tensor([0, 0, 1, 1])
    labels = Tensor(np.random.randn(16, 1), dtype=ms.float32)
    logit, extra_output = net(x, y)
    assert extra_output.shape == (2, 1)
    loss_fn = nn.MSELoss()
    def forward(x, y, labels):
        logits, extra_output = net(x, y)
        loss = loss_fn(logits, labels)
        return loss, logits, extra_output
    grad_fn = ops.value_and_grad(forward, grad_position=None,
weights=net.trainable_params(), has_aux=True)
    # raised TypeError here
    (loss, logits, extra_output), inputs_gradient = grad_fn(x, y, labels)
    assert extra_output.shape == (2, 1)

if __name__ == '__main__':
    main()
```

3. 根因分析



value_and_grad 是同时计算网络的正向输出和梯度。Graph 网络输出是一个 tuple，（logits, extra_output）

- logits 是 nn.Dense 的输出结果，类型 float32。
- extra_output 是 ops.nonzero 的输出结果，类型 int64。

梯度计算时由于 MindSpore 采用的是反向自动微分机制， 会对输出结果求和后再对输入求导。

- 当输出类型不一致的情况下就会报错。

4. 解决方案

两种方法：

- 1.把 nonzero 算子的返回值转换成 float32，转换的算子是 Cast。
- 2.使用返回值是 float32 的算子。

比如 ops.count_nonzero

- 第 1 种方案：

```
import numpy as np
import mindspore as ms
import mindspore.ops as ops
from mindspore import Tensor, nn, context
```

```
class Graph(nn.Cell):
    def __init__(self):
        super().__init__()
        self.net = nn.Dense(10, 1)
        self.cast = ops.Cast()
    def construct(self, x, y):
        logits = self.net(x)
        extra_output = ops.nonzero(y > 0)
        extra_output = self.cast(extra_output, ms.float32)
        return logits, extra_output

def main():
    context.set_context(mode=context.GRAPH_MODE)
    net = Graph()
    x = Tensor(np.random.randn(16, 10), dtype=ms.float32)
    y = Tensor([0, 0, 1, 1])
    labels = Tensor(np.random.randn(16, 1), dtype=ms.float32)
    logit, extra_output = net(x, y)
    print(extra_output.shape)
    print(extra_output)
    assert extra_output.shape == (2, 1)
    loss_fn = nn.MSELoss()
    def forward(x, y, labels):
        logits, extra_output = net(x, y)
        loss = loss_fn(logits, labels)
        return loss, logits, extra_output
    grad_fn = ops.value_and_grad(forward, grad_position=None,
weights=net.trainable_params(), has_aux=True)
    # raised TypeError here
    (loss, logits, extra_output), inputs_gradient = grad_fn(x, y, labels)
    assert extra_output.shape == (2, 1)

if __name__ == '__main__':
    main()
```

- 第 2 种方案：

```
import numpy as np
import mindspore as ms
import mindspore.ops as ops
from mindspore import Tensor, nn, context

class Graph(nn.Cell):
    def __init__(self):
        super().__init__()
        self.net = nn.Dense(10, 1)
    def construct(self, x, y):
        logits = self.net(x)
        extra_output = ops.count_nonzero(x=y, keep_dims=True, dtype=ms.float32)

        return logits, extra_output

def main():
    context.set_context(mode=context.GRAPH_MODE)
    net = Graph()
    x = Tensor(np.random.randn(16, 10), dtype=ms.float32)
    y = Tensor([0, 0, 1, 1])
```

```
    labels = Tensor(np.random.randn(16, 1), dtype=ms.float32)
    logit, extra_output = net(x, y)
    #assert extra_output.shape == (2, 1)
    loss_fn = nn.MSELoss()
    def forward(x, y, labels):
        logits, extra_output = net(x, y)
        loss = loss_fn(logits, labels)
        return loss, logits, extra_output
    grad_fn = ops.value_and_grad(forward, grad_position=None,
weights=net.trainable_params(), has_aux=True)
    # raised TypeError here
    (loss, logits, extra_output), inputs_gradient = grad_fn(x, y, labels)
    #assert extra_output.shape == (2, 1)

if __name__ == '__main__':
    main()
```

最终可以正常执行。

6.25 训练过程中评测超时导致训练过程发生中断

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend/GPU/CPU

软件环境: MindSpore 版本: 2.3.0 RC1

执行模式： 不限

Python 版本: 3.7

操作系统平台: linux

2. 报错信息

2.1 问题描述

4 机 32rank 训练过程中发生中断

```
[ERROR] DEVICE(192563,ffff90722010,python):2025-03-18-20:50:24.716.326
[mindspore/ccsrc/plugin/device/ascend/hal/device/ascend_stream_manager.cc:241]
SyncStream] Call runtime aclrtSynchronizeStreamWithTimeout error.
----------------------------------------------------
- Ascend Error Message:
----------------------------------------------------
EE9999: Inner Error!
EE9999  The error from device(chipId:3, dieId:0), serial number is 1, event wait
timeout occurred during task execution, stream_id:122, sq_id:122, task_id:790,
event_id=73,
timeout=1980.[FUNC:ProcessStarsWaitTimeoutErrorInfo][FILE:device_error_proc.cc][LIN
E:1314]
       TraceBack (most recent call last):
       Task execute failed, device_id=3, stream_id=122, task_id=790, flip_num=0,
task_type=3(EVENT_WAIT).[FUNC:GetError][FILE:stream.cc][LINE:1483]
       rtStreamSynchronizeWithTimeout execute failed, reason=[the model stream
```

```
execute failed][FUNC:FuncErrorReason][FILE:error_message_manage.cc][LINE:50]
       synchronize stream failed, runtime result =
507011[FUNC:ReportCallError][FILE:log_inner.cpp][LINE:161]
```

3. 根因分析

客户的模型在训练一定数量的 epoch 后，会进行权重文件的评测，期间各个卡的通信停止，

HCCL_EXEC_TIMEOUT 默认值 1836，如果评测时间超过 1836s，就会导致各卡的 HCCL 连接发生超时。

分析 plog 日志，可以看到每个服务器的后 7 个 rank 的 plog 日志都打印类似的错误：

```
ReportErrorInfoForModelExecuteTask:model execute error, retCode=0x91, [the model
stream execute failed].
ProcessStarsWaitTimeoutErrorInfo:The error from device(chipId:2, dieId:0), serial
number is 1, event wait timeout occurred during task execution, stream_id:122,
sq_id:122, task_id:790, event_id=73, timeout=1980.
ProcessStarsWaitTimeoutErrorInfo:report error module_type=6, module_name=EE9999
```

判断并不是某个单点的 rank 的问题导致的问题，进一步分析训练日志。

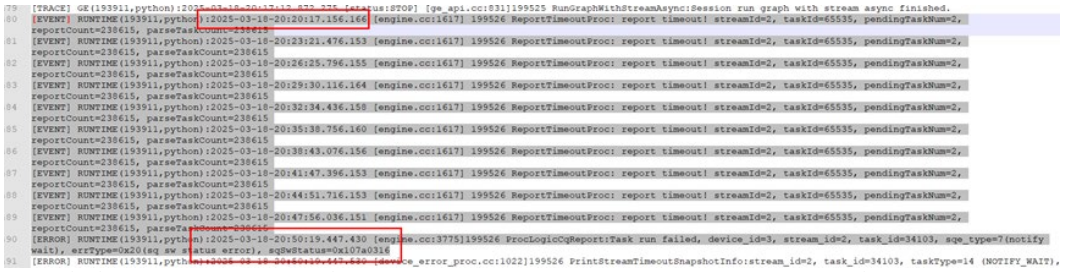
从训练日志中可以看到在第 170 个 epoch 结束后，开始进行评测任务（客户自己定制的特性）

```
INFO:root:Epoch 300/170, Step 716/716, size (640, 640), loss: 4.7977, lbox: 0.0291, lobj: 0.0031, lcls: 0.0053, cur_
INFO:root:Epoch 300/170, epoch time: 5.32 min.
INFO:root:the training process is: {'processed': 79.17, 'total': 100, 'msg': '', 'info': {}}
INFO:root:load ckpt from "./result/weights/exp10_detection-1_170.ckpt" success.
INFO:root:Test step 1/853, cost time 0.43s
INFO:root:Test step 2/853, cost time 5.15s
WARNING:root:WARNING: Batch NMS time limit 20.0s exceeded, this batch process 5/12 sample.
INFO:root:Test step 3/853, cost time 21.66s
time="2025-03-18T20:17:41+08:00" level=info msg="analyzer is diagnosing" file="analyzer.go:308" Command=analyze Comp
time="2025-03-18T20:17:42+08:00" level=info msg="time required for conclude:  0 ms" file="analyzer.go:416" Command=a
WARNING:root:WARNING: Batch NMS time limit 20.0s exceeded, this batch process 8/12 sample.
INFO:root:Test step 4/853, cost time 23.62s
INFO:root:Test step 5/853, cost time 13.40s
INFO:root:Test step 6/853, cost time 0.32s
INFO:root:Test step 7/853, cost time 0.31s
INFO:root:Test step 8/853, cost time 0.32s
INFO:root:Test step 9/853, cost time 0.33s
INFO:root:Test step 10/853, cost time 19.70s
time="2025-03-18T20:18:50+08:00" level=info msg="record a new fault code: A050954" file="node_fault_code.go:93" Comm
time="2025-03-18T20:18:50+08:00" level=info msg="a fault code A050954 detected by maos-node-agent" file="listener.go
INFO:root:Test step 11/853, cost time 14.41s
```

然后评测未完全完成时发生错误

```
INFO:root:Test step 158/853, cost time 23.73s
time="2025-03-18T20:48:50+08:00" level=info msg="record a new fault code: A050954" file="node_fault_code.go:93" Command
time="2025-03-18T20:48:50+08:00" level=info msg="a fault code A050954 detected by maos-node-agent" file="listener.go:71
WARNING:root:WARNING: Batch NMS time limit 20.0s exceeded, this batch process 5/12 sample.
INFO:root:Test step 159/853, cost time 25.16s
WARNING:root:WARNING: Batch NMS time limit 20.0s exceeded, this batch process 11/12 sample.
INFO:root:Test step 160/853, cost time 20.57s
WARNING:root:WARNING: Batch NMS time limit 20.0s exceeded, this batch process 5/12 sample.
INFO:root:Test step 161/853, cost time 23.40s
WARNING:root:WARNING: Batch NMS time limit 20.0s exceeded, this batch process 5/12 sample.
INFO:root:Test step 162/853, cost time 20.92s
[ERROR] DEVICE(192555,fffe51f4b160,python):2025-03-18-20:50:22.140.619 [mindspore/ccsrc/plugin/device/ascend/hal/device
[ERROR] DEVICE(192555,fffe51f4b160,python):2025-03-18-20:50:22.140.716 [mindspore/ccsrc/plugin/device/ascend/hal/device
[ERROR] DEVICE(192563,fffe5ef45160,python):2025-03-18-20:50:22.292.665 [mindspore/ccsrc/plugin/device/ascend/hal/device
```

从评测开始到发生错误，时间差不多是 30 分钟。



刚好在在 plog 报错前的日志对应的时间段可以看到下面的记录，也是在首次打印 ReportTimeoutProc: report timeout!的地方后 30 分钟发生 notify wait 的 timeout 错误。

这个时间也与下面错误信息中的 timeout=1980 基本吻合

```
EE9999  The error from device(chipId:3, dieId:0), serial number is 1, event wait
timeout occurred during task execution, stream_id:122, sq_id:122, task_id:790,
event_id=73,
timeout=1980.[FUNC:ProcessStarsWaitTimeoutErrorInfo][FILE:device_error_proc.cc][LIN
E:1314]
```

Notify wait 的错误是因为 HCCL 超时，超时时间是通过 HCCL_CONNECT_TIMEOUT 和 HCCL_EXEC_TIMEOUT 来控制的。检查运行环境的 HCCL_CONNECT_TIMEOUT 和 HCCL_EXEC_TIMEOUT 配置，运行环境变量中并没有 HCCL_CONNECT_TIMEOUT(默认时间 60)和 HCCL_EXEC_TIMEOUT(默认时间 1800（秒）)。

基于上面的分析，原因判断为评测时间超过了 HCCL 的超时时间。

4. 解决方案

配置 HCCL_CONNECT_TIMEOUT 和 HCCL_EXEC_TIMEOUT 参数，增加超时时间，减少因为评测任务执行时间长导致的超时失败。

6.26 昇腾 910 上 CodeLlama 推理报错 get fail deviceLogicId[0]

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend910

软件环境: MindSpore 版本: 2.2.10

执行模式： 不限

Python 版本: 3.8.15

操作系统平台: linux

2. 报错信息

报错信息

```
[ERROR] HCCL(2073294, python3) :2023-12-22-10:36:04.666.308 [op_base.cc:56]
[2082087] [HcclGetDeviceId] get fail deviceLogicId[0]
[ERROR] HCCL(2073294, python3) :2023-12-22-10:36:04.938.806 [op_base.cc:56]
```

```
[2082083] [HcclGetDeviceId] get fail deviceLogicId[0]
[ERROR] HCCL(2073294, python3) :2023-12-22-10:36:05.015.145 [op_base.cc:56]
[2082079] [HcclGetDeviceId] get fail deviceLogicId[0]
[ERROR] HCCL(2073294, python3) :2023-12-22-10:36:05.054.705 [op_base.cc:56]
[2082088] [HcclGetDeviceId] get fail deviceLogicId[0]
[ERROR] HCCL(2073294, python3) :2023-12-22-10:36:05.098.737 [op_base.cc:56]
[2082082] [HcclGetDeviceId] get fail deviceLogicId[0]
[ERROR] HCCL(2073294, python3) :2023-12-22-10:36:05.151.236 [op_base.cc:56]
[2082084] [HcclGetDeviceId] get fail deviceLogicId[0]
[ERROR] HCCL(2073294, python3) :2023-12-22-10:36:05.307.610 [op_base.cc:56]
[2082080] [HcclGetDeviceId] get fail deviceLogicId[0]
[ERROR] HCCL(2073294, python3) :2023-12-22-10:36:05.505.706 [op_base.cc:56]
[2082086] [HcclGetDeviceId] get fail deviceLogicId[0]
[ERROR] HCCL(2073294, python3) :2023-12-22-10:36:05.554.217 [op_base.cc:56]
[2082077] [HcclGetDeviceId] get fail deviceLogicId[0]
[ERROR] HCCL(2073294, python3) :2023-12-22-10:36:05.615.027 [op_base.cc:56]
[2082076] [HcclGetDeviceId] get fail deviceLogicId[0]
[ERROR] HCCL(2073294, python3) :2023-12-22-10:36:05.859.348 [op_base.cc:56]
[2082081] [HcclGetDeviceId] get fail deviceLogicId[0]
[ERROR] HCCL(2073294, python3) :2023-12-22-10:36:06.917.314 [op_base.cc:56]
[2082078] [HcclGetDeviceId] get fail deviceLogicId[0]
```

3. 根因分析

HCCL 初始化接口未调用。

4. 解决方案

检查推理配置脚本，增加 rank_table_file

```
[ascend_context]
#plugin_custom_ops=KVcache
# enable_custom_op=All
provider=ge
rank_table_file=/data/pangshiguan/mindformer-hk/hccl_8p_01234567_7.213.219.27.json
```

6.27 昇腾 910 上 CodeLlama 导出 mindir 模型报错 rankTablePath is invalid

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend910

软件环境: MindSpore 版本: 2.2.10

执行模式： 不限

Python 版本: 3.8.15

操作系统平台: linux

2. 报错信息

报错 rankTablePath is invalid

```
Ascend Error Message:
I0003: In [HcomInitByFile],value [/user/config/nbstart_hccl.json] for parameter
[rankTablePath]is invalid. Reason: The collec
as an invalid argument. Reason[/user/config/nbstart_hccl.json]
    Solution: Try again with a validargument.
    TraceBack (most recent calllast):
    PluginManager InvokeAll failed.(FUNC:Initialize]
(FILE:ops_kernel_manager.cc][LINE:96]
   OpsManager initialize failed. [FUNC:InnerInitialize] [FILE:gelib. cc] [LINE:237]
    GELib::InnerInitialize failed.[FUNC:Initialize][FILE:gelib.cc][LINE:165]
Please search "Ascend Error Message"at https://w.mindspore.cn for errorcode
description)
C++ Call Stack: (For framework developers
```

3. 根因分析

未生成 RANK_TABLE_FILE

4. 解决方案

运行 mindformers/tools/hccl_tools.py 生成 RANK_TABLE_FILE 的 json 文件

```
export PATH=/usr/local/Ascend/driver/tools/:${PATH}
# 运行如下命令，生成当前机器的 RANK_TABLE_FILE 的 json 文件
python ./mindformers/tools/hccl_tools.py --device_num "[0,8)"
```

配置环境变量 export RANK_TABLE_FILE=xxx.json

6.28 权重文件被异常修改导致加载权重提示 Failed to read the checkpoint file

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend/GPU/CPU

软件环境: MindSpore 版本: 2.5.0

执行模式： 不限

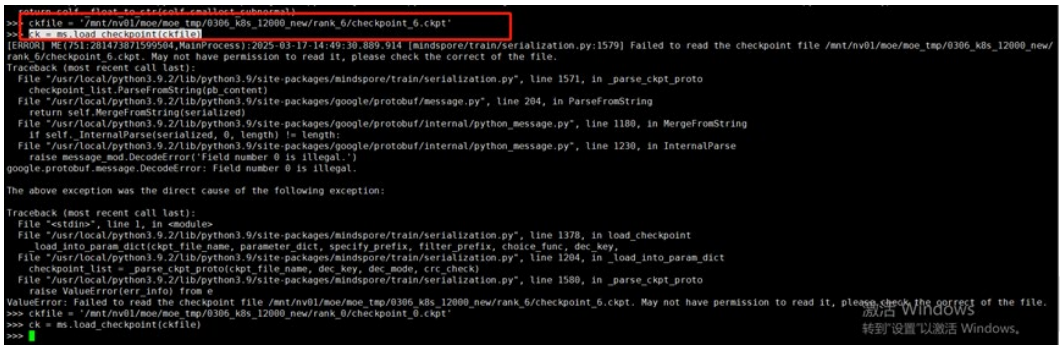
Python 版本: 3.7.6

操作系统平台: linux

2. 报错信息

2.1 问题描述

使用部分 MindSpore 2.5.0 版本加载权重文件时，提示：



2.2 脚本信息

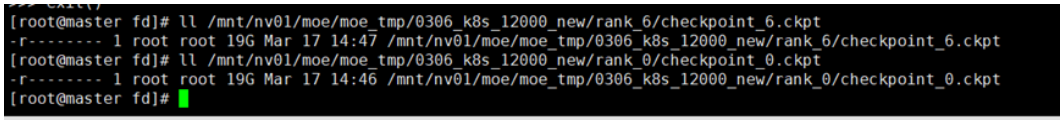
```
import mindspore as ms
ckpt_file = 'xxxx'
ckpt = ms.load_checkpoint(ckpt_file)
```

2.3 报错信息

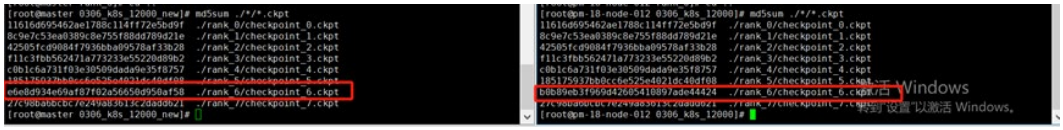
```
Failed to read the checkpoint file xxxxx. May not have permission to read it,
please check the correct of the file
```

3. 根因分析

对比能正常加载的权重和不能正常加载的权重的权限，都是 400，所以并不是权限的问题。



怀疑是文件自身的问题，一共 8 个权重文件，通过 md5sum 对比现在加载的权重和原来保存出来的权重，发现加载失败的权重文件 md5 码不匹配。



重新拷贝上传该文件后，问题解决。

4. 解决方案

重新拷贝上传异常的权重文件，确保 md5 码能匹配后再进行操作。

6.29 Mindformers 模型启动时因为 host 侧 OOM 导致任务被 kill

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.5.0、MindFormers 版本：1.3.0

执行模式： 不限

Python 版本: 3.7.6

操作系统平台: linux

2. 报错信息

2.1 问题描述

基于 Mindformers 进行 Telechat2 35B 模型 SFT 训练，模型拉起训练任务时进行权重自动转换阶段失败，提示

2.2 报错信息

```
[ERROR] TBE Subprocess(task distribute] raise error[], main process disappeared!
```

3. 根因分析

参考错误提示，并没有 CANN 或者 mindformers 侧相关的报错，提示信息表示进程是强行结束的。

报错是发生在权重转换阶段，客户提供的权重文件是使用 64 卡训练出来的分布式权重文件，并进行了合并，合并后的 ckpt 权重大小约到 130G。

当前需要使用 8 卡来跑训练，预期每个卡的权重大小是 23G 左右，客户开启了权重自动转换，mindformers 拉起模型时的 yaml 里面配置为：

```
# 打开权重自动转换开关
auto_trans_ckpt: True
```

自动转换时，每个卡都会单独加载 ckpt 权重文件，相当于 host 侧需要把 8 个 ckpt 文件都加载 host 侧的内存里面，合计需要约 1T 的内存空间。

重新拉起一次任务，并通过 top 命令监控内存的使用信息，发现内存的 free 值逐渐减少，最后可用内存不足，然后进程被 kill 掉，问题复现。

针对这种打大权重文件需要进行转换的场景，建议使用离线转换方式：

https://gitee.com/mindspore/mindformers/blob/r1.3.0/docs/feature_cards/Transform_Ckpt.md

然后 mindformers 使用的 yaml 文件直接使用转换后的权重，并将自动转换关闭

```
# 打开权重自动转换开关
auto_trans_ckpt: False
```

重新拉起模型，本次 host 侧没有因为内存耗尽导致触发 kill 进程，任务顺利拉起。

权重加载时，需要将权重文件从服务器硬盘上读到 host 的内存中，然后再从 host 内存下发到 NPU 的显存，在权重文件从硬盘往 host 内存中加载时，服务器侧内存不足导致操作系统将进程 kill。

4. 解决方案

需要对大的权重文件进行转换时，需要评估 host 的可用内存是否足够，可用 free -g 来检查可用内存大小。如果内存不够，建议采用离线转换方式：

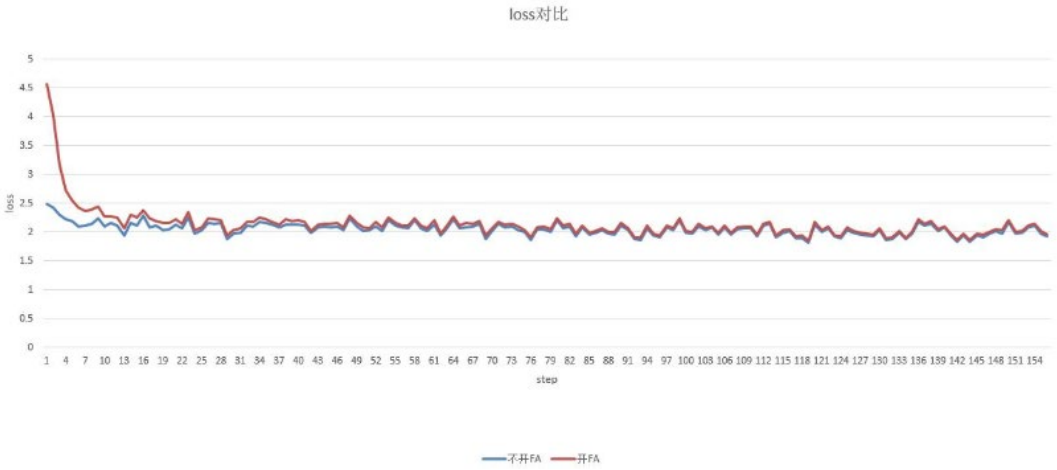
https://gitee.com/mindspore/mindformers/blob/r1.3.0/docs/feature_cards/Transform_Ckpt.md

然后 mindformers 使用的 yaml 文件直接使用转换后的权重，并将自动转换关闭。

6.30 昇腾 910FlashAttention 适配 alibi 问题

1. 问题现象

昇腾 910 上不开 FA 和开 FA 后 loss 对不齐。如下图中的两个曲线，前几个 step loss 相差较大。



2. 根因分析

1.没有开启 alibi 参数

2.开启了 alibi 参数，但是因为海思的 pse 定义跟 mindspore 的不同，所以在送到海思前要先把 scale 除掉。

海思实现方法：

```
score = (k * v + pse) * scale
```

mindspore 实现方法：

```
score = k * v * scale + pse
```

4. 解决方案

目前除以 scale 的操作已经封装在 nn.FlashAttention 算子中，可以直接使用。(ms whl 包中的路径 mindspore/nn/layer/flash_attention.py)

如果使用的 mindspore 版本比较旧，还没有封装 nn.FlashAttention 算子，可用下面方法规避，只针对 mindformers： 方法 1. 修改类 mindformers.modules.layers.AlibiTensorV2 的__init__方法

```
# self.slopes = Tensor(slopes[None, :, None, None],mstype.float32) # (num_heads, 1)
self.slopes = Tensor(slopes[None, :, None, None] * math.sqrt(head_dim),
mstype.float32) # (num_heads, 1)
```

方法 2.修改 mindformers.modules.layers.build_alibi_tensor_v2 方法

```
# slopes = np.expand_dims(np.expand_dims(slopes, 1), 1)
slopes = np.expand_dims(np.expand_dims(slopes, 1), 1) * math.sqrt(head_dim)
```

6.31 llama3.1-8b 的 lora 微调，不开启权重转换会导致维度不匹配，开启了之后会报错找不到 rank1 的 ckpt，但是 strategy 目录里面是全的

1. 报错信息

```
The checkpoint file of rank1 is needed for converting rank1's checkpoint, but it is missing.
```

2. 解决方案

此问题需要检查训练使用的 yaml 配置文件；

yaml 配置文件中设置 auto_trans_ckpt=False,enable_parallel_optimizer=False 后，模型微调可以运行起来了

6.32 Mindspore+MIndformer 训练 plog 报错 halMemAlloc failed，drvRetCode=6

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.4.0

执行模式： 不限

Python 版本: 3.7.6

操作系统平台: linux

2. 报错信息

2.1 问题描述

训练日志报错,plog 报错 halMemAlloc failed，drvRetCode=6

2.2 报错信息

3. 根因分析

1. 训练日志报错“MindSpore dataset is terminated with err msg: Exception thrown from dataset pipeline. Refer to 'Dataset Pipeline Error Message'. Tdt Send data failed. The details refer to 'Ascend Error Message'. ” 从训练日志看不出具体的报错原因。
2. 查看 plog 日志，通过 grep -rn "ERROR" ./ |sort -t ":" -k 4,6 |head 获取到首报错信息。
3. 错误信息“halMemAlloc failed，drvRetCode=6”说明 HCCL 通信显存申请失败。
4. 查看训练 yaml 配置参数 max_device_memory 的值。NPU 总可用显存-max_device_memory=HCCL 通信可用显存。 因此逐步降低 max_device_memory 的值来增大 HCCL 通信可使用的显存。

HCCL 通信 HBM 不足导致。

4. 解决方案

方法 1： 逐步降低 max_device_memory，再次拉起训练。

方法 2： 通过 export HCCL_BUFFSIZE=xxx 设置环境变量的值降低通信域内存占用。例如 export HCCL_BUFFSIZE=100

每个 HCCL 通信组会默认占用 200MB 的内存，那么在通信组较多的场景下，就容易出现设备侧内存不足的报错。解决方法是设置 HCCL_BUFFSIZE 环境变量修改通信域内存占用。

6.33 MindSpore 的离线权重转换接口说明及转换过程

1. 权重相关接口说明

权重转换主要涉及 6 个接口 ckpt_to_safetensors、safetensors_to_ckpt、merge_pipeline_strategys 、transform_checkpoint、unified_safetensors 和 load_distributed_checkpoin

接口	使用链接	功能说明

ckpt_to_safetensors	https://www.mindspore.cn/docs/zh-CN/r2.4.0/api_python/mindspore/mindspore.ckpt_to_safetensors.html?highlight=ckpt_to_safetensors#mindspore.ckpt_to_safetensors	权重格式从 ckpt 转到 safetensors 的转换 分布式权重和完整都可以
safetensors_to_ckpt	https://www.mindspore.cn/docs/zh-CN/r2.4.0/api_python/mindspore/mindspore.safetensors_to_ckpt.html?highlight=safetensors_to_ckpt#mindspore.safetensors_to_ckpt	权重格式从 safetensors 到 ckpt 的转换 分布式权重和完整都可以
transform_checkpoint	https://www.mindspore.cn/docs/zh-CN/master/api_python/mindspore/mindspore.transform_checkpoints.html?highlight=transform_checkpoints#mindspore.transform_checkpoints	由源切分策略转换到目标切分策略, 输入是 ckpt 输出只能是 ckpt 输入是 safetensors，输出可以 ckpt 或 safetensors 入参必须是带 rank 的，完整权重放到 rank0 下 建议所有 rank 大小小于 100G
unified_safetensors	https://www.mindspore.cn/docs/zh-CN/r2.4.0/api_python/mindspore/mindspore.unified_safetensors.html?highlight=unified_safetensors#mindspore.unified_safetensors	将分布式 safetensors 文件合并为一份完整分片的 safetensors 文件。
load_distributed_checkpoint	https://www.mindspore.cn/docs/zh-CN/r2.4.0/api_python/mindspore/mindspore.load_distributed_checkpoint.html?highlight=load_distributed_checkpoint#mindspore.load_distributed_checkpoint	将完整 safetensors 权重转换为分布式或完整的 ckpt 或 safetensors 权重 输入：格式 safetensors 输出：2.4.0 版本输出格式只能是 safetensors，2.4.0 之后可以是 ckpt 或 safetensors，默认 'safetensors'
merge_pipeline_strategys	https://www.mindspore.cn/docs/zh-CN/r2.4.0/api_python/mindspore/mindspore.merge_pipeline_strategys.html?highlight=merge_pipeline_strategys#mindspore.merge_pipeline_	策略文件合并

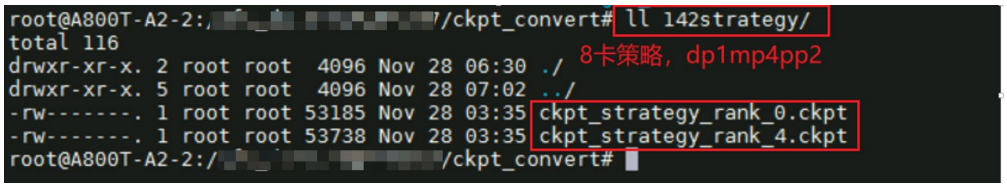
	strategys	

1. merge_pipeline_strategys：合并分布式策略文件

```
import mindspore as ms
ms.merge_pipeline_strategys(
src_strategy_dirs="源策略文件所在的目录",
dst_strategy_file="合并后的策略文件名")
```

src_strategy_dirs：源策略文件,目录下面放了所有的策略文件,不需要按照 rank 存。

dst_strategy_file：合并后的策略文件,指定到策略文件名称，例如 /xxx/xxx/merge.ckpt



2. ckpt_to_safetensors：把 ckpt 转换为 safetensors，分布式或完整 ckpt 都可以

```
import mindspore as ms
ms.ckpt_to_safetensors(
file_path="源 ckpt 文件的目录或文件",
save_path="转换为 safetensors 文件的目录",
processes_num=进程数)
```

file_path：源 ckpt 权重

分布式权重：每个 rank 下面一个 ckpt,指定到 rank 的上一层路径。

完整权重：指定到权重文件。

save_path：转换为 safetensors 文件的目录，输出和输入的存储路径一样

processes_num：整数,可取 4,8,16,32 等,根据自己机器内存、cpu 使用情况决定

分布式权重存储样例：

```
roo                                  /ckpt_convert/weight_o# ls -lR 142distributed_ckpt/
142distributed_ckpt/:
total 32
drwx------. 2 root root 4096 Nov 28 06:39 rank_0
drwx------. 2 root root 4096 Nov 28 06:39 rank_1
drwx------. 2 root root 4096 Nov 28 06:40 rank_2
drwx------. 2 root root 4096 Nov 28 06:40 rank_3
drwx------. 2 root root 4096 Nov 28 06:39 rank_4
drwx------. 2 root root 4096 Nov 28 06:39 rank_5
drwx------. 2 root root 4096 Nov 28 06:40 rank_6
drwx------. 2 root root 4096 Nov 28 06:40 rank_7

142distributed_ckpt/rank_0:
total 1645332
-r--------. 1 root root 1684819265 Nov 28 06:39 dst_ckpt_prefix0.ckpt

142distributed_ckpt/rank_1:
total 1645332
-r--------. 1 root root 1684819265 Nov 28 06:39 dst_ckpt_prefix1.ckpt

142distributed_ckpt/rank_2:
total 1645332
-r--------. 1 root root 1684819265 Nov 28 06:40 dst_ckpt_prefix2.ckpt

142distributed_ckpt/rank_3:
total 1645332
-r--------. 1 root root 1684819265 Nov 28 06:40 dst_ckpt_prefix3.ckpt

142distributed_ckpt/rank_4:
total 1645340
-r--------. 1 root root 1684827568 Nov 28 06:39 dst_ckpt_prefix4.ckpt

142distributed_ckpt/rank_5:
total 1645340
-r--------. 1 root root 1684827568 Nov 28 06:39 dst_ckpt_prefix5.ckpt

142distributed_ckpt/rank_6:
total 1645340
-r--------. 1 root root 1684827568 Nov 28 06:40 dst_ckpt_prefix6.ckpt

142distributed_ckpt/rank_7:
total 1645340
-r--------. 1 root root 1684827568 Nov 28 06:40 dst_ckpt_prefix7.ckpt
root@                                /ckpt_convert/weight_o#
```

3. safetensors_to_ckpt：把 safetensors 转换为 ckpt，分布式或完整 safetensors 都可以

```
import mindspore as ms
ms.safetensors_to_ckpt(
file_path="safetensors 权重文件或目录 ",
save_path="转换为 ckpt 文件的目录",
processes_num=进程数)
```

file_path：源权重，包含 safetensors 文件的目录路径或单个 safetensors 文件(.safetensors) 的路径。备注：指定目录时必须按照 rank 存储

save_path：转换为 ckpt 文件的目录

processes_num：整数,可取 4,8,16,32 等,根据自己机器内存、cpu 使用情况决定

4. transform_checkpoint: 由源切分策略转换到目标切分策略

```
import mindspore as ms
ms.transform_checkpoints (src_checkpoints_dir, dst_checkpoints_dir, ckpt_prefix,
src_strategy_file=None, dst_strategy_file=None, process_num=1,
output_format='ckpt')
print("Transform ckpt DONE", flush=True)
```

src_checkpoints_dir:源权重的路径,分布式权重指定到 rank 的上一层路径,完整权重指定到文件名

dst_checkpoints_dir:目标权重的路径

ckpt_prefix：目标权重文件名的前缀

src_strategy_file：源权重合并后的策略文件,对于完整权重指定 None

dst_strategy_file：目标权重合并后的策略文件,对于完整权重指定 None

process_num：线程格数,根据自己机器 cpu、内存使用情况绝对

output_format："ckpt" 或"safetensors"

5. unified_safetensors：将分布式 safetensors 文件合并为一份完整分片的 safetensors 文件

```
import mindspore as ms
ms.unified_safetensors(
src_dir="safetensors 格式的文件目录",
src_strategy_file="源权重的切分策略(需合并)",
dst_dir="生成文件的目录")
```

src_dir: safetensors 分布式权重的目录，指定到 rank 的上一层路径

src_strategy_file：源权重的切分策略(需合并) ，即 merge_pipeline_strategys 的输出结果

dst_dir：生成文件的目录

6. load_distributed_checkpoint：将完整 safetensors 权重转换为分布式或完整的 ckpt 或 safetensors 权重

```
import mindspore as ms
ms.load_distributed_checkpoint(
network=None, #可以不是 None，直接传入 Net
predict_strategy="目标权重的策略文件(合并后)", #完整权重指定 None
format='safetensors', #取值 safetensors
output_format='safetensors', #默认是'safetensors'，取值 ckpt 或 safetensors，2.4.0 之后可以使用 ckpt
unified_safetensors_dir="离线合并后的目录(unified_safetensors 的结果)",
dst_safetensors_dir="生成目标权重目录" )
```

2. 权重转换原则

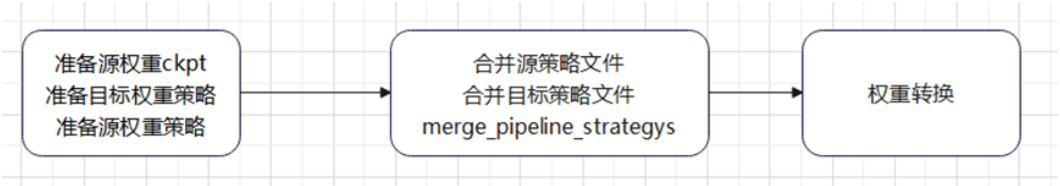
原则一：权重是完整权重的，策略文件指定 None

原则二： 策略文件和权重格式没有关系，同样策略，ckpt 和 safetensors 可以共用策略文件。

原则三：分布式权重传 rank 上一层路径，完整权重传文件路径

原则四：对于权重所有 rank 加起来比较大的，建议都先转 safetensors

3. 离线权重转换过程



以源 A 集群迁移到目标 B 集群为例：

第一步：获取所需要的相关文件,共需要以下三类文件

1. 源集群 A 的切分策略信息：`netA.strategy`
2. 源集群 A 的权重,每个 rank 的 ckpt 文件：`rank_3/netA_10_6.ckpt`

备注：每个进程生成一个目录,每个目录下有不同阶段的多个 ckpt 文件,上面的文件命名表示：这是第`3`个 rank,在第`10`个 epoch、第`6`个 step 训练结束时生成的 ckpt 文件

1. 目标集群 B 的切分策略信息：`netB.strategy`

第二步：合并所有 rank 下的策略信息为一个完整的策略文件

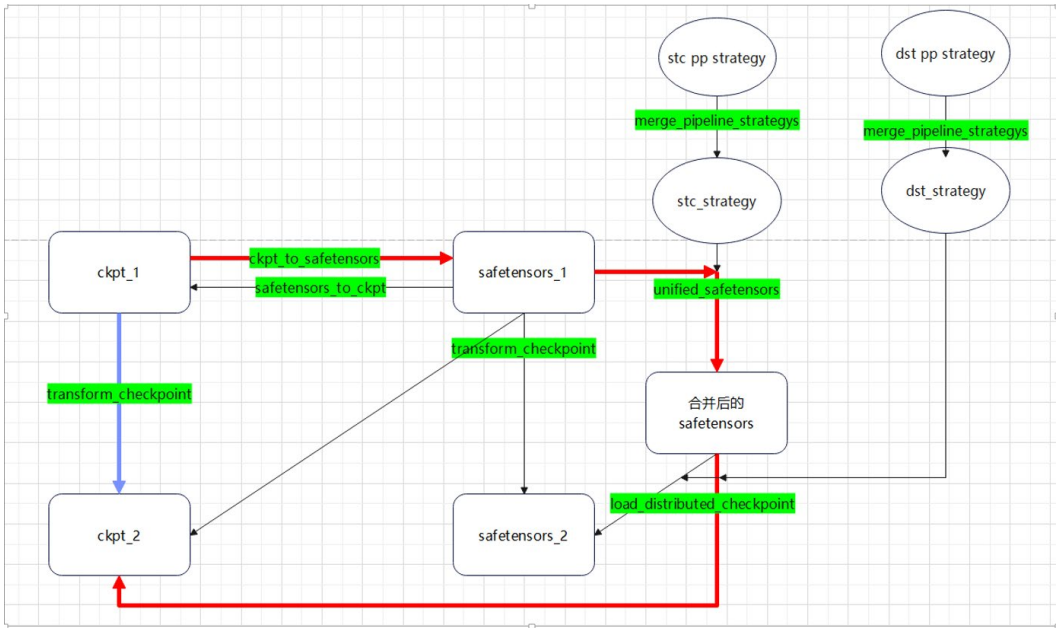
1. 默认情况,每张卡都会生成一个策略文件,但这个策略文件只表示这个 pipeline 下的切分策略,如果想得到集群完整的切分策略,需要把这些策略文件合并一下(源、目标都要合并)。使用 ms.merge_pipeline_stratrge 方法把他们合并成一个完整的策略

第三步： 权重转换

有了第一步的三类文件,并进行了第二步的策略文件合并后,就可以进行权重转换了。

使用 ms.transform_checkpoint 方法,输入权重文件(目录)和集群 A、集群 B 的策略文件,就可以输出集群 B 的权重文件了。

或者使用 unified_safetensors ms.load_distributed_checkpoint 进行权重转换



分布式 ckp1 转分布式 ckp2：推荐红色流，权重不大可以使用蓝色流

相关案例：

场景	相关案例链接
场景一：safetensors 转换为 ckpt 权重	https://www.hiascend.com/forum/thread-0211117099514819214l-1-1.html
场景二：ckpt 转换为 safetensors 权重	https://www.hiascend.com/forum/thread-0211117099514819214l-1-1.html
场景三：ckpt 分布式权重 A 转换为其他分布式权重 B	https://www.hiascend.com/forum/thread-0215170999041958157-1-1.html
场景四：完整 ckpt 权重转换为分布式权重	https://www.hiascend.com/forum/thread-0211117100871680814l-1-1.html
场景五：分布式权重转换为完整 ckpt 权重	https://www.hiascend.com/forum/thread-0211117100871680814l-1-1.html

6.34 MindSpore 的 ckpt 格式完整权重和分布式权重互转

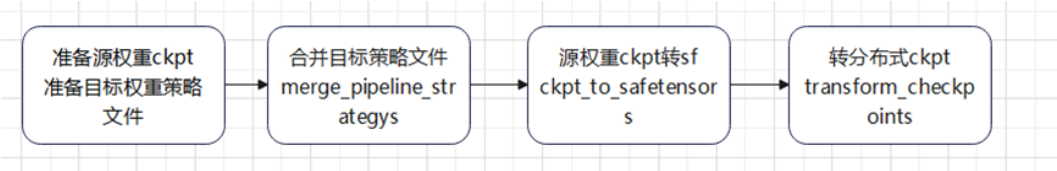
1. 场景描述

完整 ckpt 权重转换为分布式权重

2. 具体实现

用 safetensors 格式转换

方法一：通过 ckpt_to_safetensors + transform_checkpoint



建议使用 safetensors 进行权重转换

```
import mindspore as ms
ms.transform_checkpoints (src_checkpoints_dir, dst_checkpoints_dir, ckpt_prefix,
src_strategy_file=None, dst_strategy_file=None, process_num=1, output_format='ckpt')
print("Transform ckpt DONE", flush=True)
```

src_checkpoints_dir:权重 A 的路径,分布式权重指定到 rank 的上一层路径,完整权重指定到文件名

dst_checkpoints_dir:生成权重 B 的路径

ckpt_prefix：生成权重 B 文件名的前缀

src_strategy_file：权重 A 合并后的策略文件,对于完整权重指定 None

dst_strategy_file：权重 B 合并后的策略文件,对于完整权重指定 None

process_num：线程格数,根据自己机器 cpu、内存使用情况绝对

output_format："ckpt" 或"safetensors"

样例：

1. 合并目标策略文件

```
root@[redacted]/ckpt_convert# cat merge_pipeline_strategys.py
import mindspore as ms
ms.merge_pipeline_strategys(
src_strategy_dirs="/[redacted]/ckpt_convert/142strategy/",
dst_strategy_file="/[redacted]/ckpt_convert/142merge.ckpt")
root@[redacted]/ckpt_convert# ll -l
total 152
drwxr-xr-x.  5 root root   4096 Nov 28 06:30 ./
drwxr-xr-x. 34 root root  12288 Nov 28 06:20 ../
-rw-r--r--.  1 root root 109249 Nov 28 06:30 142merge.ckpt
drwxr-xr-x.  2 root root   4096 Nov 28 06:30 142strategy/
drwxr-xr-x.  2 root root   4096 Nov 28 05:01 144strategy/
-rw-r--r--.  1 root root    206 Nov 27 14:17 ckpt_to_sf.py
-rw-r--r--.  1 root root   4435 Nov 27 06:45 hf_sf_to_cpkt.py
-rw-r--r--.  1 root root    327 Nov 27 07:39 load_distributed_checkpoint.py
-rw-r--r--.  1 root root    187 Nov 28 06:30 merge_pipeline_strategys.py
-rw-r--r--.  1 root root    197 Nov 27 08:17 sf_to_ckpt.py
drwxr-xr-x.  6 root root   4096 Nov 27 14:21 weight_o/
root@[redacted]/ckpt_convert#
```

2. ckpt 转 safetensors

```
import mindspore as ms
ms.ckpt_to_safetensors(
file_path="/xxx/ckpt_convert/weight_o/net.ckpt",
save_path="/xxx/ckpt_convert/weight_o/ckpt_to_sf_all/",
processes_num=8)
```

输入:

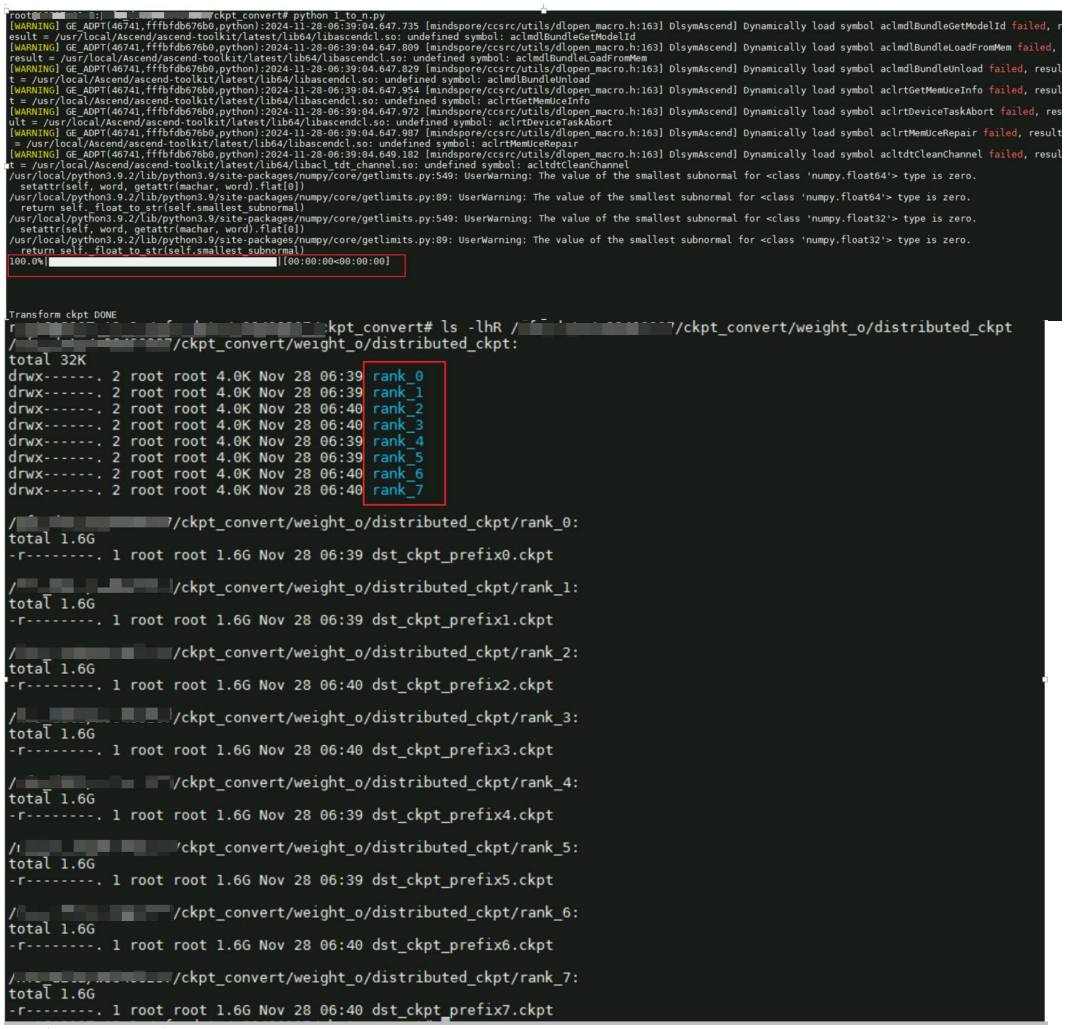
```
root@[redacted]/ckpt_convert# ll /[redacted]/ckpt_convert/weight_o/net.ckpt
-r--------. 1 root root 13476860573 Nov 27 14:09 /[redacted]/ckpt_convert/weight_o/net.ckpt
root@[redacted]/ckpt_convert#
```

输出：

```
root@[redacted]/ckpt_convert# ll /[redacted]/ckpt_convert/weight_o/ckpt_to_sf_all/
total 13161020
drwxr-xr-x. 2 root root        4096 Nov 27 14:11 ./
drwxr-xr-x. 6 root root        4096 Nov 27 14:05 ../
-rw-r--r--. 1 root root 13476875608 Nov 27 14:13 net.safetensors
root@A800T-A2-2:[redacted]/ckpt_convert#
```

3. 调用 transform_checkpoints 进行 safetensors 转分布式 ckpt

```
import mindspore as ms
ms.transform_checkpoints
("/xxx/ckpt_convert/weight_o/ckpt_to_sf_all/rank_0/net.safetensors",
"/xxx/ckpt_convert/weight_o/distributed_ckpt", "dst_ckpt_prefix",
src_strategy_file=None, dst_strategy_file="/xxx/ckpt_convert/142merge.ckpt",
process_num=4, output_format='ckpt')
print("Transform ckpt DONE", flush=True)
```



方法二：通过 ckpt_to_safetensors + load_distributed_checkpoint

```
import mindspore as ms
ms.load_distributed_checkpoint(
network=None, #可以不是 None，直接传入 Net
predict_strategy="权重 B 的策略文件(合并后)", #完整权重指定 None
format='safetensors', #默认是 ckpt，取值 ckpt 或 safetensors
output_format='ckpt', #默认是'safetensors'，取值 ckpt 或 safetensors，2.4.0 之后可以使用
unified_safetensors_dir="离线合并后的目录(上一步产生的)",
dst_safetensors_dir="生成的权重 B 目录" )
```

样例：

1. 合并目标策略文件

```
root@ /ckpt_convert# cat merge_pipeline_strategys.py
import mindspore as ms
ms.merge_pipeline_strategys(
src_strategy_dirs="/ /ckpt_convert/142strategy/",
dst_strategy_file="/ /ckpt_convert/142merge.ckpt")
root@ /ckpt_convert# ll -l
total 152
drwxr-xr-x.  5 root root   4096 Nov 28 06:30 ./
drwxr-xr-x. 34 root root  12288 Nov 28 06:20 ../
-rw-r--r--.  1 root root 109249 Nov 28 06:30 142merge.ckpt
drwxr-xr-x.  2 root root   4096 Nov 28 06:30 142strategy/
drwxr-xr-x.  2 root root   4096 Nov 28 05:01 144strategy/
-rw-r--r--.  1 root root    206 Nov 27 14:17 ckpt_to_sf.py
-rw-r--r--.  1 root root   4435 Nov 27 06:45 hf_sf_to_cpkt.py
-rw-r--r--.  1 root root    327 Nov 27 07:39 load_distributed_checkpoint.py
-rw-r--r--.  1 root root    187 Nov 28 06:30 merge_pipeline_strategys.py
-rw-r--r--.  1 root root    197 Nov 27 08:17 sf_to_ckpt.py
drwxr-xr-x.  6 root root   4096 Nov 27 14:21 weight_o/
root@ /ckpt_convert#
```

2. ckpt 转 safetensors

```
import mindspore as ms
ms.ckpt_to_safetensors(
file_path="/xxx/ckpt_convert/weight_o/net.ckpt",
save_path="/xxx/ckpt_convert/weight_o/ckpt_to_sf_all/",
processes_num=8)
```

输入：

```
root@A800T-A2-2:/ /ckpt_convert# ll /  ckpt_convert/weight_o/net.ckpt
-r--------. 1 root root 13476860573 Nov 27 14:09 /  /ckpt_convert/weight_o/net.ckpt
root@A800T-A2-2:/  7/ckpt_convert#
```

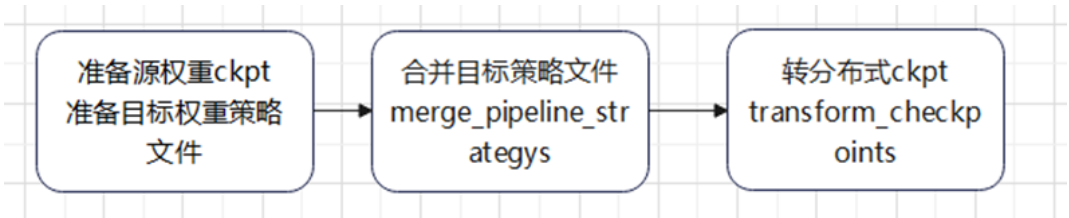
输出：

```
root@ /ckpt_convert# ll /  /ckpt_convert/weight_o/ckpt_to_sf_all/
total 13161020
drwxr-xr-x. 2 root root        4096 Nov 27 14:11 ./
drwxr-xr-x. 6 root root        4096 Nov 27 14:05 ../
-rw-r--r--. 1 root root 13476875608 Nov 27 14:13 net.safetensors
root@ /ckpt_convert#
```

3. 调用 load_distributed_checkpoint 进行 safetensors 转分布式 ckpt

```
import mindspore as ms
ms.load_distributed_checkpoint(
network=None,
checkpoint_filenames=None,
predict_strategy="/xxx/ckpt_convert/142merge.ckpt",
format='safetensors',
unified_safetensors_dir="/xxx/ckpt_convert/weight_o/ckpt_to_sf_all/rank_0/net.safet
ensors",
dst_safetensors_dir="/xxx/ckpt_convert/weight_o/142distributed_ckpt_1" )
```

方法三：transform_checkpoint—不推荐



transform_checkpoints 会对输入格式自动识别

```
import mindspore as ms
ms.transform_checkpoints
("/xxx/ckpt_convert/weight_o/ckpt_to_sf_all/rank_0/net.ckpt",
"/xxx/ckpt_convert/weight_o/distributed_ckpt", "dst_ckpt_prefix",
```

```
src_strategy_file=None, dst_strategy_file="/xxx/ckpt_convert/142merge.ckpt",
process_num=4, output_format='ckpt')
print("Transform ckpt DONE", flush=True)
```

分布式权重转换为完整 ckpt 权重

用 safetensors 进行格式转换

方法一：通过 ckpt_to_safetensors +transform_checkpoint

```
import mindspore as ms
ms.transform_checkpoints (src_checkpoints_dir, dst_checkpoints_dir, ckpt_prefix,
src_strategy_file=None, dst_strategy_file=None, process_num=1, output_format='ckpt')
print("Transform ckpt DONE", flush=True)
```

src_checkpoints_dir:权重 A 的路径,分布式权重指定到 rank 的上一层路径,完整权重指定到文件名

dst_checkpoints_dir:生成权重 B 的路径

ckpt_prefix：生成权重 B 文件名的前缀

src_strategy_file：权重 A 合并后的策略文件,对于完整权重指定 None

dst_strategy_file：权重 B 合并后的策略文件,对于完整权重指定 None

process_num：线程格数,根据自己机器 cpu、内存使用情况决定

output_format："ckpt" 或"safetensors"

样例：

```
import mindspore as ms
ms.transform_checkpoints ("/xxx/ckpt_convert/weight_o/ distributed_ckpt ",
"/xxx/ckpt_convert/weight_o/all_ckpt", "dst_ckpt_prefix",
src_strategy_file="src_strategy", dst_strategy_file=None, process_num=4,
output_format='ckpt')
print("Transform ckpt DONE", flush=True)
```

方法二：通过 ckpt_to_safetensors+ unified_safetensors+ load_distributed_checkpoint

ckpt_to_safetensors+ unified_safetensors---得到分片的 sf，可以转分片的 ckpt，然后手动合并。

6.35 MindSpore+MindFormer-r.1.2.0 微调 qwen1.5 报错

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend910

软件环境: MindSpore 版本: 2.3.0、mindformer=r.1.2.0

执行模式： 不限

Python 版本: 3.8

操作系统平台: linux

2. 报错信息

2.1 问题描述

使用 ms2.3.0，mindformer-r.1.2.0，微调 qwen1_5_7b,配置文件如下：

2.2 配置信息

```
seed: 42
output_dir: './output'
load_checkpoint: ''
auto_trans_ckpt: False  # If true, auto transform load_checkpoint to load in
distributed model
only_save_strategy: False
resume_training: False
run_mode: 'finetune'
# trainer config
trainer:
  type: CausalLanguageModelingTrainer
  model_name: 'qwen2_7b'
# if True, do evaluate during the training process. if false, do nothing.
# note that the task trainer should support _evaluate_in_training function.
do_eval: False
eval_step_interval: -1       # num of step intervals between each eval, -1 means no
step end eval.
eval_epoch_interval: 50       # num of epoch intervals between each eval, 1 means
eval on every epoch end.

# runner config
runner_config:
  epochs: 5
  batch_size: 1
  sink_mode: True
  sink_size: 1

# wrapper cell config
runner_wrapper:
  type: MFTrainOneStepCell
  scale_sense:
    type: DynamicLossScaleUpdateCell
    loss_scale_value: 4096
    scale_factor: 2
    scale_window: 1000
  use_clip_grad: True

# optimizer
optimizer:
  type: FP32StateAdamWeightDecay
  beta1: 0.9
  beta2: 0.95
  eps: 1.e-8
  learning_rate: 1.e-6
  weight_decay: 0.01
```

```
# lr schedule
lr_schedule:
  type: CosineWithWarmUpLR
  learning_rate: 1.e-6
  warmup_ratio: 0.01
  total_steps: -1 # -1 means it will load the total steps of the dataset

# dataset
train_dataset: &train_dataset
  data_loader:
    type: MindDataset
    dataset_dir: ""
    shuffle: True
  input_columns: ["input_ids", "target_ids", "attention_mask"]
  num_parallel_workers: 8
  python_multiprocessing: False
  drop_remainder: True
  batch_size: 1
  repeat: 1
  numa_enable: False
  prefetch_size: 1
train_dataset_task:
  type: CausalLanguageModelDataset
  dataset_config: *train_dataset

# eval dataset
eval_dataset: &eval_dataset
  data_loader:
    type: MindDataset
    dataset_dir: ""
    shuffle: False
  input_columns: ["input_ids", "target_ids", "attention_mask"]
  num_parallel_workers: 8
  python_multiprocessing: False
  drop_remainder: False
  repeat: 1
  numa_enable: False
  prefetch_size: 1
eval_dataset_task:
  type: CausalLanguageModelDataset
  dataset_config: *eval_dataset

use_parallel: True
# parallel context config
parallel:
  parallel_mode: 1 # 0-data parallel, 1-semi-auto parallel, 2-auto parallel, 3-
hybrid parallel
  gradients_mean: False
  enable_alltoall: False
  full_batch: True
  search_mode: "sharding_propagation"
  enable_parallel_optimizer: True
  strategy_ckpt_save_file: "./ckpt_strategy.ckpt"
  parallel_optimizer_config:
```

```
    gradient_accumulation_shard: False
    parallel_optimizer_threshold: 64

# default parallel of device num = 8 910
parallel_config:
  data_parallel: 1
  model_parallel: 8
  pipeline_stage: 1
  use_seq_parallel: True
  micro_batch_num: 1
  vocab_emb_dp: True
  gradient_aggregation_group: 8
# when model parallel is greater than 1, we can set micro_batch_interleave_num=2,
that may accelerate the train process.
micro_batch_interleave_num: 1

# recompute config
recompute_config:
  recompute: True
  select_recompute: False
  parallel_optimizer_comm_recompute: False
  mp_comm_recompute: False
  recompute_slice_activation: False

# callbacks
callbacks:
  - type: MFLossMonitor
  - type: CheckpointMonitor
    prefix: "qwen2"
    save_checkpoint_steps: 5000
    keep_checkpoint_max: 1
    integrated_save: False
    async_save: False
  - type: ObsMonitor

# mindspore context init config
context:
  mode: 0 #0--Graph Mode; 1--Pynative Mode
  device_target: "Ascend"
  enable_graph_kernel: False
  max_call_depth: 10000
  max_device_memory: "55GB"
  save_graphs: False
  save_graphs_path: "./graph"
  device_id: 0
  jit_config:
    jit_level: "O0"
  ascend_config:
    precision_mode: "must_keep_origin_dtype"

# model config
model:
  model_config:
    type: LlamaConfig
    batch_size: 1 # add for increase predict
```

```
    seq_length: 256
    hidden_size: 4096
    num_layers: 32
    num_heads: 32
    vocab_size: 151936
    intermediate_size: 11008
    qkv_has_bias: True
    rms_norm_eps: 1.0e-6
    theta: 1000000.0
    max_position_embedding: 32768
    emb_dropout_prob: 0.0
    eos_token_id: 151643
    pad_token_id: 151643
    compute_dtype: "bfloat16"
    layernorm_compute_type: "float32"
    softmax_compute_type: "float16"
    rotary_dtype: "float16"
    param_init_type: "float32"
    use_past: False
    extend_method: "None" # support "None", "PI", "NTK"
    use_flash_attention: True
    fine_grain_interleave: 1
    qkv_concat: False
    block_size: 32
    num_blocks: 128
    offset: 0
    checkpoint_name_or_path: ""
    repetition_penalty: 1
    max_decode_length: 512
    top_k: 0
    top_p: 0.8
    do_sample: False
    compute_in_2d: True
    # configuration items copied from Qwen
    pet_config:
      pet_type: lora
      # configuration of lora
      lora_rank: 64
      lora_alpha: 16
      lora_dropout: 0.05
      target_modules: '.*wq|.*wk|.*wv|.*wo|.*w1|.*w2|.*w3'
      freeze_exclude: ["*wte*", "*lm_head*"]

    rotary_pct: 1.0
    rotary_emb_base: 1000000
    kv_channels: 128

  arch:
    type: LlamaForCausalLM

processor:
  return_tensors: ms
  tokenizer:
    model_max_length: 4096
    vocab_file: "/path/vocab.json"
```

```
    merges_file: "/path/merges.txt"
    unk_token: "<|endoftext|>"
    eos_token: "<|endoftext|>"
    pad_token: "<|endoftext|>"
    type: Qwen2Tokenizer
   type: Qwen2Processor

processor:
  return_tensors: ms
  tokenizer:
    model_max_length: 4096
    vocab_file: "/path/vocab.json"
    merges_file: "/path/merges.txt"
    unk_token: "<|endoftext|>"
    eos_token: "<|endoftext|>"
    pad_token: "<|endoftext|>"
    type: Qwen2Tokenizer
   type: Qwen2Processor

# metric
metric:
  type: PerplexityMetric

eval_callbacks:
  - type: ObsMonitor

auto_tune: False
filepath_prefix: './autotune'
autotune_per_step: 10

profile: False
profile_start_step: 1
profile_stop_step: 10
init_start_profile: False
profile_communication: False
profile_memory: True
layer_scale: False
layer_decay: 0.65
lr_scale_factor: 256

# aicc
remote_save_url: "Please input obs url on AICC platform."复制
```

2.3 报错信息

```
"trainer': {'model_name' :'qwen2_7b','type': 'CausalLanguageModelingTrainer'},
"use parallel': False}
2024-10-09 14:09:44,272 - mindfo mers[mindfommers/trainer/base trainer.py:787] -
INFO -   .Model Compiling, Please Wait a Moment
[WARNING] ME(1739195:281472916631776, MainProcess):2024-10-09- 14:09:44.273.000
[mindspo re/train/model.py:1328] For MFLossMonitor callback, f'step_begin',
'step.end', 'epoch_end', 'epoch_begin'} methods may not be supported in later
version, Use methods prefixed with 'on_train' or 'on_eval' instead when using
customized callbacks-
[WARNING] ME(1739195:281472916631776,MainProcess) :2024-10-09-14:09:44.273.000
[mindspore/train/model .py:1328] For Local20bsMonitor callback, {'epoch_end',
```

```
'steend'} methods may not be supported in later version, Use methods prefixed with
"on train' or 'on eval' instead when using customized callbacks.(ERRORJ DEVICE
(1739195,ffff8535b0e0.python) :2024-10-09- 14:12:37.081.576
[mindspore/ccsrc/plugin/device/ascend/hal/device/ascend_vmm_adapter.cc:206]
AllocDeviceMem] Reserve memory address failed.
[ERROR] PRE_ACT (1739195, ffff8535b0e0,python) :2024-10-09-14: 12:37.081.722
[mindspore/ccs rc/backend/common/mem_reuse/mem_dynamic_allocator.cc:414]
AddMemBlockAnd
MemBufByEagerFree] AllocDeviceMemByEagerFree failed, alloc size : 0.  [ERRORI
RUNTIME FRAMEWORK(1739195, ffff8535b0e0,python) :2024-10-09- 14:12:37.081.857
[mindspore/ccsrc/runtime/graph_scheduler/actor/data_prepare_actor.cc:129]
SyncTensorData] #umsg#Memory not enough:#umsg#Device( id:0) memory isn't enough and
alloc failed, kernel name: Default/data-895, alloc size: 2B.
IERRORI DEVICE(1739195, ffff8535b0e0.python) :2024-10-09-14:12:37.082 -022
[mindspore/ccsrc/plugin/device/ascend/hal/device/ascend_vmm_adapter.cc:206]
AllocDeviceMem] Reserve memory address failed.
TERROR] PRE ACT( 1739195, ffff8535b0e0, python) :2024-10-09-14:12:37.082.057
[mindspore/ccsrc/backend/common/mem_ reuse/mem dynamic_allocato r.cc:414]
AddMemBlockAndMemBufByEagerFree] AllocDeviceMemByEagerFree failed, alloc size : 0.
(ERROR]'RUNTIME FRAMEWORK(1739195, ffff8535b0e0,python) :2024-10-09-14:
12:37.082.100 [mindspore/ccsrc/runtime/graph scheduler/actor/data prepare
actor.cc:129] SyncTensorData] #umsg#Memory not enough:#umsg#Devićе(id:0) memory
isn't enough and alloc failed, kernel name: Default/data-1360, alloc size: 16B.
(ERRORI DEVICE(1739195, ffff8535b0e0python) :2024-10-09-14:12:37.082.239
[mindspore/ccsrc/plugin/device/ascend/hal/device/ascend_wmm_adapter .cc:206]
AllocDeviceMem] Reserve memory address failed.
[ERROR] PRE_ ACT (1739195, ffff8535b0e0,python) :2024-10-09-14: 12:37.082.273
[mindspore/ccs rc/backend/common/mem_reuse/mem_dynamic_allocator.cc:414]
AddMemBlockAndMemBufByEagerFree] AllocDeviceMemByEagerFree failed, alloc size : 0.
(ERROR] RUNT IME_FRAMEWORK(1739195, ffff8535b0e0,python) :2024-10-09-14:
12:37.082.313
[mindspore/ccsrc/runtime/graph_scheduler/actor/data_prepare_actor.cc:129]
SyncTensorData] #umsg#Memory not enough:#umsg#Device(id:0) memory isn't enough and
alloc failed, kernel name: Default/data-1658, alloc size: 2B.
(ERROR] DEVICE(1739195, ffff8535b0e0,python) :2024-10-09-14: 12:37.082.424
[mindspore/ccsrc/plugin/device/ascend/hal/device/ascend_vmm_adapter.cc:206]
AllocDeviceMem] Reserve memory address failed.
[ERROR] PRE_ACT(1739195, ffff8535b0e0,python) :2024-10-09-14: 12:37.082.457
[mindspore/ccsrc/backend/common/mem_reuse/mem_dynamic_allocator.cc:414]
AddMemBlockAndMemBufByEagerFree] AllocDeviceMemByEagerFree failed, alloc size : 0.
[ERROR] RUNT IME_FRAMEWORK(1739195, ffff8535b0e0,python) :2024-10-09-14:
12:37.082.519
[mindspore/ccsrc/runtime/graph_scheduler/actor/data_prepare_actor.cc:129]
SyncTensorData] #umsg#Memory not enough:#umsg#Device( id:0) memory isn't enough and
alloc failed, kernel name: Default/data-862, alloc size: 2B.
[ERROR] DEVICE (1739195, ffff8535b0e0,python) :2024-10-09-14: 12:37.082.648
Imindspore/ccsrc/plugin/device/ascend/hal/device/ascend_vmm_adapter.cc:206]
AllocDeviceMem] Reserve memory address failed.
[ERROR] PRE ACT(1739195, ffff8535b0e0, python): :2024-10-09-14:12:37.082.679
[mindspore/ccsrc/backend/common/mem_reuse/mem_dynamic_allocato r.cc:414]
AddMemBlockAndMemBufByEagerFree] AllocDeviceMemByEagerFree failed, alloc size : 0.
(ERROR] RUNTIME FRAMEWORK(1739195, ffff8535b0e0,python) :2024-10-09-14:
12:37.082.719 [mindspore/ccsrc/runtime/graph scheduler/actor/data prepare
actor.cc:129] SyncTensorData] #umsg#Memory not enough:#umsg#Device( id:0) memory
isn't enough and alloc failed, kernel name: Default/data-1339,- alloc size: 16B.
```

```
[ERROR] DEVICE (1739195,ffff8535b0e0,python) :2024-10-09-14:12:37.082.832 [mindspo
re/ccsrc/plugin/device/ascend/hal/device/ascend_vmm_adapter .cc:206] AllocDeviceMem]
Reserve memory address failed.
[ERROR] PRE. ACT(1739195, ffff8535b0e0,python) :2024-10-09-14: 12:37.082.863
[mindspore/ccs rc/backend/common/mem_reuse/mem_dynamic_allocator.cc:414]
AddMemBlockAndMemBufByEagerFree] AllocDeviceMemByEagerFree failed, alloc size
```

3. 根因分析

报错是

```
AllocDeviceMem] Reserve memory address failed.
```

配置文件中

```
max_device_memory: "55GB"
parallel_config:
  data_parallel: 1
  model_parallel: 8
  pipeline_stage: 1
  use_seq_parallel: True
  micro_batch_num: 1
  vocab_emb_dp: True
  gradient_aggregation_group: 8
```

也就是要求 64G npu ram，8 卡。当前环境并不满足，因此报错。

4. 解决方案

因为配置文件配置的是 8 卡 64G，可以直接更换环境。

如果当前是 8 卡 32G 环境，可以修改 max_device_memory 为 27G 进行尝试。

```
max_device_memory: "27GB"
```

如果还是报内存不足的话可以修改如下参数，减小 seq_length，num_layers，max_decode_length 等参数。

```
seq_length: 256
   hidden_size: 4096
   num_layers: 32
   num_heads: 32
   max_decode_length: 512
```

6.36 Mindformer 训练 plog 中算子 GatherV2_xxx_high_precision_xx 报错

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.4.0

执行模式： 不限

Python 版本: 3.7.6

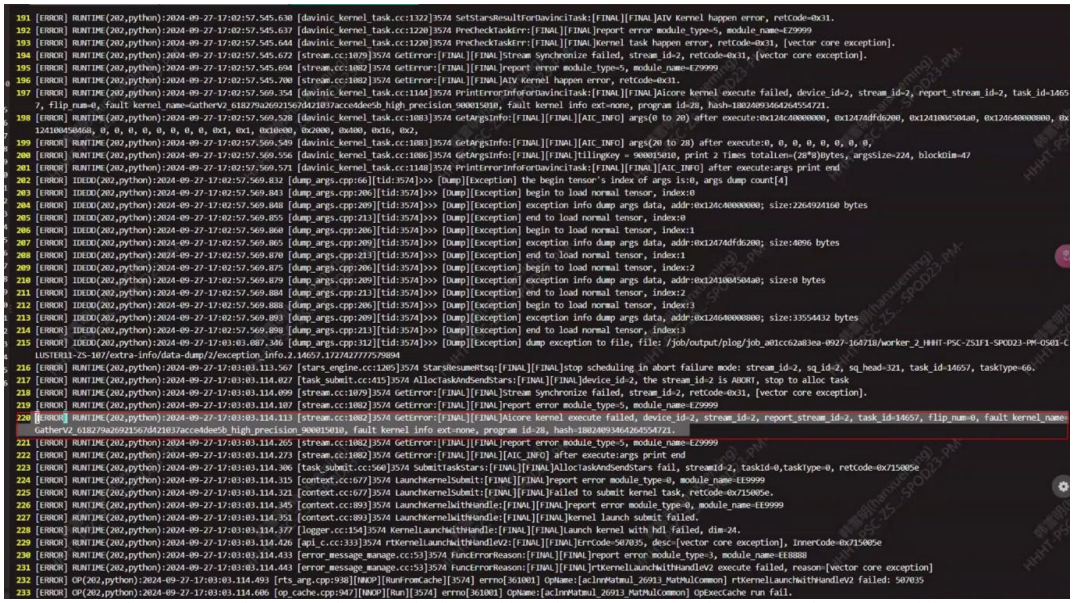
操作系统平台: linux

2. 报错信息

训练日志报错,plog 报错 GatherV2_xxx_high_precision_xx

3. 根因分析

训练日志报错但看不出具体的报错原因。 查看 plog 日志，发现算子 GatherV2_xxx_high_precision_xx 报错。



GatherV2 算子报错在 plog 中，一般都是数据集的词表和模型词表大小不一样，导致的越界。

4. 解决方案

查看 yaml 中 vocab_size 和数据集使用的词表大小是否匹配。

查看词表大小的方法：

如果使用 sentencepiece 库：

```
import sentencepiece as sp

sp_model = sp.SentencePieceProcessor()
sp_model.load("tokenizer.model")
vocab_size = sp_model.get_piece_size()
print(f"词表大小 (vocab_size): {vocab_size}")
```

如果使用 transformers 库中的相关类（如果适用的话）：

```
from transformers import LlamaTokenizer

tokenizer = LlamaTokenizer.from_pretrained("tokenizer.model")
vocab_size = tokenizer.vocab_size
print(f"词表大小 (vocab_size): {vocab_size}")
```

对于使用 MIndFormer

```
from mindformers import LlamaTokenizer

tokenizer = LlamaTokenizer.from_pretrained("tokenizer.model")
vocab_size = tokenizer.vocab_size
print(f"词表大小 (vocab_size): {vocab_size}")
```

不同的库和方法可能会有一些细微的差异，具体使用哪种方法取决于你项目中已安装和使用的库以及具体的需求。如果在读取过程中遇到权限问题，确保你有足够的权限访问该文件。

6.37 mindformers 进行 Lora 微调后的权重合并

1. mindformers 当前 Lora 微调保存权重，lora 权重未合并，不支持直接离线推理

```
model.layers.37.attention.wq.mindpet_delta_lora_a
model.layers.37.attention.wq.mindpet_delta_lora_b
```

2. LoRA 低秩适配原理

根据 lora 实现原理，只需要将原本的权重加上 sacle 低秩的权重就可以得到模型最终的权重。

3. 脚本实现 LoRA 权重合并

```
def transpose(weight, fan_in_fan_out):
    return weight.T if fan_in_fan_out else weight

ms.transform_checkpoints(src_ckpt_dir, dst_ckpt_dir, prefix, src_ckpt_strategy,
dst_ckpt_strategy)
print("......Over transform......")
print("......Start transform lora......")
#          lora 超参 lora_scaling 计算 lora_r=lora_rank
lora_scaling = args.lora_alpha * args.lora_r
save_path = dst_ckpt_dir + "/rank_0/" + prefix + "0.ckpt"
# such as   model.layers.attention.wq.weight
#          model.layers.attention.wq.mindpet_delta_lora_a
#          model.layers.attention.wq.mindpet_delta_lora_b
param_dict = ms.load_checkpoint(save_path)
lora_keys = [k for k in param_dict if 'lora_a' in k]
non_lora_keys = [k for k in param_dict if not 'lora_' in k]
param_dict_lora = {}
for k in non_lora_keys:
    param_dict_lora[k] = param_dict[k].clone()
for k in lora_keys:
    print(f"merging {k}")
    original_key = k.replace('_lora_a', '').replace('mindpet_delta', 'weight')
    assert original_key in param_dict
    lora_a_key = k
    lora_b_key = k.replace('lora_a', 'lora_b')

    param_dict_lora[original_key] = Parameter(param_dict_lora[original_key].value()
+ (
          transpose(param_dict[lora_b_key].value() @ param_dict[lora_a_key].value(),
                  False) * lora_scaling), name=original_key
```

```
                                        )
print("......Start save transform lora......")
os.remove(save_path)
save_path = dst_ckpt_dir + "/rank_0/" + prefix + "_lora_merged.ckpt"
ms.save_checkpoint(param_dict_lora, save_path)
```

6.38 pangu-100b 2k 集群线性度问题定位

1. 环境信息

硬件环境(Ascend/GPU/CPU): Ascend910B

软件环境: MindSpore 版本: 2.2.0

执行模式： Graph

2. 报错信息

pangu-100b 在不同规模的集群上的性能：64->30s/step, 1k->33s/step, 2k->45s/step，在2k 集群上性能劣化，不满足线性度。

3. 根因分析

profile 后分析 timeline 数据，发现有个 allgather 算子通信很慢，占比很大。

存图后定位到 allgather 算子是 attention_mask 产生的，attention_mask 的shape[batch_size，sql_length，sql_length]，在数据集里生成会导致 oom，所以采用入图的方式生成。

attention_mask 初始化脚本如下：

```
def construct(self, input_ids, input_position, attention_mask, lables,loss_mask):
    r"""Forward process of the pangu model
    input_ids: [b, seq_lenght]
    input_position: [b, seq_lenght]
    attention_mask: [b, seq_lenght, seq_lenght]
    labels: [b, lenght]
    loss_mask: [b, seq_lenght]
    """
    attention_mask = self.tril(F.ones((self.batch_size, self.seq_lenght,
self.seq_lenght), dtype=matype.float16))
```

根本原因是 tril 算子不支持 shard，配置了切分策略不生效，所以会插入 allgather 通信算子，本身 shape 比较大，导致通信耗时长，拖慢整个训练。

4. 解决方案

采用 np 方式初始化 Tensor 后，消除掉了 allgather 通信算子，问题解决，2k 集群性能32s/step，线性度达标。

```
self.attention_mask= Tensor(np.tril(xxx))
```

6.39 Mixtral 8*7B 大模型精度问题总结

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend910

软件环境: MindSpore 版本: 2.3

执行模式(PyNative/Graph)： Graph

2. 精度问题与分析

2.1 正向 rotary embedding 精度不一致

使用 tensordump 进行网络正向对比，发现 rotary embedding 处精度不一致，对比源码后发现 rotary embedding 的 base 参数不一致，修改后精度一致。

2.2 正向 MOE 部分精度不一致

对比 moe 层，发现 torch 的 q、k、v 等 tensor 的排布和 ms 的不同，无法对比，只能对比整体的输入和输出，发现误差放大到十分位。对比脚本，发现 torch moe 中容量为全容量设置与 mindspore 不一致，修改 ms 为专家全容量后，前向最终 loss 误差为万分位。

2.3 adam 优化器精度不一致

正向对比完成后，对比反向精度，发现反向 loss 误差在百分位，torch 的 fuseAdam 优化器与 ms 的 FP32AdamW 实现不同引入误差。分析 fuseAdam 源码，使用 ms 的 Adam 可与 torch 对齐。ms 使用 Adam 优化器与权重 FP32 初始化已与 torch 对齐，整体误差千分位。权重 fp16/bf16 初始化，和 torch 还是无法对齐，存在误差，同时 ms 当前不支持权重精度低于梯度的精度，导致报错。

6.40 大模型内存占用调优

模型运行时如果 host 内存不断增长，则可能发生了内存泄漏。发生内存泄漏的位置可能是数据集读取部分和网络部分，可以分别跑数据集和网络来定位内存泄漏的位置。

内存分析工具：memory_profiler

memory_profiler 是 python 的第三方库，其功能是逐行代码分析内存占用，可以通过装饰器或命令行的方式使用。具体使用方式可以参

host 内存溢出的场景

报错信息如下：

LLVM ERROR: out of memory

Memory consumption is more than …, which may cause oom error.

dataset/util/arena.cc … cudaHostAlloc failed, ret[2], out of memory

出现类似的错误信息，则说明 host 内存溢出，需要优化内存占用。

调优技巧

可通过以下 checklist 进行数据处理的内存占用调优：

数据集操作 调整方式

预取 set_prefetch_size 配置预取

数据加载类（**Dataset）、batch、map 操作的工作线程数 num_parallel_workers 减少工作线程数，但会影响数据读取性能

map 操作 减少/合并 map 操作

shuffle 操作 减少 buffer_size 大小

batch 操作 减少 batch_size 大小

6.41 大模型网络算法参数和输出对比

网络参数对比

网络参数（这里指网络权重）是网络的基础，如果网络参数与标杆网络不一致，后续的对比将没有意义。

排查参数个数是否一致；

排查参数 shape 是否一致；

排查参数是否有冻结；

如果排查网络参数一致后，可以使用权重迁移工具，将两个的网络参数都初始化相同的值，便于进行下述对比。

网络输出对比

控制相同网络权重和输入，对比网络的输出结果（包含正向网络输出和 loss）是否一致。如果一致，说明网络基本没有问题，可以优先进行其他对比。如果不一致，可以用下面的逐层对比找出差异产生的具体位置。

注意：对比时需要去除网络中的随机元素。

逐层对比

静态图

如果 MindSpore 网络运行在静态图模式下，可以使用 Dump 工具进行对比。

Dump 对比：可以导出所有算子的输入输出，与标干网络对应算子的输入输出进行对比。

工具链接：Dump 功能调试

动态图

如果网络运行在动态图模式下，可以使用 Debug、USEFUL_TOOLS 或 Hook 功能进行对比。

Debug 对比：使用 pycharm、vscode 等工具的 Debug 功能，在指定位置打断点，查看算子的输入输出。

使用 USEFUL_TOOLS 工具对比：如果 MindSpore 网络与目标网络完全一致，包括节点的名称，个数等。此时，可以使用 USEFUL_TOOLS 工具自动对比 pytorch 或 mindspore 网络中的同名节点的正向与反向的输入与输出。工具链接：USEFUL_TOOLS

使用 Hook 功能对比：Hook 功能可以捕获中间层算子的输入、输出数据以及反向梯度。工具链接：Hook 功能

6.42 使用 jit 编译加速时编译流程中的常见 if 控制流问题

if 语句产生的控制流

Type Join Failed

在前端编译的推理阶段，会对节点的抽象类型(包含 type 、shape 等)进行推导，常见抽象类型包括 AbstractScalar 、AbstractTensor 、AbstractFunction 、AbstractTuple 、AbstractList 等。在一些场景比如多分支场景，会对不同分支返回值的抽象类型进行 join 合并，推导出返回结果的抽象类型。如果抽象类型不匹配，或者 type /shape 不一致，则会抛出以上异常。

当出现类似 Type Join Failed: dtype1 = Float32, dtype2 = Float16 的报错时，说明数据类型不一致，导致抽象类型合并失败。根据提供的数据类型和代码行信息，可以快速定位出错范围。此外，报错信息中提供了具体的抽象类型信息、节点信息。代码样例如下：

```
import numpy as np
import mindspore as ms
import mindspore.ops as ops
from mindspore import nn, jit

class Net(nn.Cell):
    def __init__(self):
        super().__init__()
        self.relu = ops.ReLU()
        self.cast = ops.Cast()

    @jit
    def construct(self, x, a, b):
        if a > b:    # if 的两个分支返回值的 type 不一致
            return self.relu(x)    # shape: (2, 3, 4, 5), dtype:Float32
        else:
            return self.cast(self.relu(x), ms.float16)    # shape: (2, 3, 4, 5),
dtype:Float16

input_x = ms.Tensor(np.random.rand(2, 3, 4, 5).astype(np.float32))
input_a = ms.Tensor(2, ms.float32)
input_b = ms.Tensor(6, ms.float32)
net = Net()
out_me = net(input_x, input_a, input_b)
```

执行结果如下：

```
TypeError: Cannot join the return values of different branches, perhaps you need to
make them equal.
```

```
Type Join Failed: dtype1 = Float32, dtype2 = Float16.
For more details, please refer to
https://www.mindspore.cn/search?inputValue=Type%20Join%20Failed

----------------------------------------------------
- Framework Error Message: (For framework developers)
----------------------------------------------------
The abstract type of the return value of the current branch is:
AbstractTensor(shape: (2, 3, 4, 5), element: AbstractScalar(Type: Float16, Value:
ValueAny, Shape: NoShape), value_ptr: 0x15694f0, value: ValueAny),
 and that of the previous branch is:
AbstractTensor(shape: (2, 3, 4, 5), element: AbstractScalar(Type: Float32, Value:
ValueAny, Shape: NoShape), value_ptr: 0x15694f0, value: ValueAny).
The node is @1___main___Net_construct_47:CNode_45{[0]:
@1___main___Net_construct_47:CNode_42{[0]: ValueNode<Primitive> Switch, [1]:
CNode_31, [2]: ValueNode<FuncGraph> 3_✓__main___Net_construct_43, [3]:
ValueNode<FuncGraph> 4_✗__main___Net_construct_44}}, true branch:
3_✓__main___Net_construct_43
In file test.py:15, 12~31
          return self.relu(x)    # shape: (2, 3, 4, 5), dtype:Float32
          ^~~~~~~~~~~~~~~~~~

, false branch: 4_✗__main___Net_construct_44
In file test.py:17, 12~54
          return self.cast(self.relu(x), ms.float16)    # shape: (2, 3, 4, 5),
dtype:Float16
          ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


----------------------------------------------------
- C++ Call Stack: (For framework developers)
----------------------------------------------------
mindspore/core/abstract/abstract_value.cc:607 ThrowException

----------------------------------------------------
- The Traceback of Net Construct Code:
----------------------------------------------------

# In file test.py:12~17, 4~54
   @jit

# In file test.py:14~17, 8~54
       if a > b:    # if 的两个分支返回值的 type 不一致
       ^
```

通过报错信息可以得知，return self.relu(x)的节点抽象类型和 return self.cast(self.relu(x), ms.float16)的节点抽象类型不一致，故无法推导出控制流的抽象类型。如想将该段代码用图编译加速的方式提升性能，需要先修改代码，使控制流的两个分支的抽象保持一致。对于上方用例，修改方式可以是去掉 cast。

6.43 jit 编译加速功能开启时，如何避免多次重新编译

使用 jit 接口将一个函数或者 cell 用静态图编译加速时，如果输入发生以下变化，将会在每次执行时重新遍历，大大降低执行性能。因此为了避免该情况，可以通过改变脚本的方式使图编译流程识别到输入的动态状态。

Tensor 的 shape 发生改变

当 cell 或者函数输入的 tensor 的 shape 改变时，多次调用该函数或 cell 将会导致多次编译。如以下代码

```
import numpy as np
import mindspore as ms
from mindspore import nn, jit

class Net(nn.Cell):
    def __init__(self):
        super().__init__()

    @jit
    def construct(self, x, a):
        return x + a

net = Net()
out_me = []
for i in range(1, 10):
    input_x = ms.Tensor(np.random.rand(i, 3, 4, 5).astype(np.float32))
    input_a = ms.Tensor(np.random.rand(i, 3, 4, 5).astype(np.float32))
    out_me.append(net(input_x, input_a))
```

其中 net 的每次调用之间输入的 shape 都不一样，会导致每次调用 net 都重新编图。如以下代码用@jit(dynamic=1)代替@jit 可以避免多次编图，只重新编译两次。

```
import numpy as np
import mindspore as ms
from mindspore import nn, jit

class Net(nn.Cell):
    def __init__(self):
        super().__init__()

    @jit(dynamic=1)
    def construct(self, x, a):
        return x + a

net = Net()
out_me = []
for i in range(1, 10):
    input_x = ms.Tensor(np.random.rand(i, 3, 4, 5).astype(np.float32))
    input_a = ms.Tensor(np.random.rand(i, 3, 4, 5).astype(np.float32))
    out_me.append(net(input_x, input_a))
```

标量值发生改变

当 cell 或者函数的输入有标量，且标量值发生改变时，多次调用该函数或 cell 将会导致多次编译。如以下代码

```
import numpy as np
import mindspore as ms
from mindspore import nn, jit
class Net(nn.Cell):
    def __init__(self):
        super().__init__()

    @jit(dynamic=1)
    def construct(self, x, a, i):
        if i > 5:
            return x + a
        else:
            return x

net = Net()
out_me = []
for i in range(1, 10):
    input_x = ms.Tensor(np.random.rand(2, 3, 4, 5).astype(np.float32))
    input_a = ms.Tensor(np.random.rand(2, 3, 4, 5).astype(np.float32))
    out_me.append(net(input_x, input_a, i))
```

其中 net 的每次调用之间输入的标量 i 的值都不一样，会导致每次调用 net 都重新编图。如以下代码使用 mutable 接口，用 num = mutable(i)代替 i 可以避免多次编图。

```
import numpy as np
import mindspore as ms
from mindspore import nn, jit, mutable

class Net(nn.Cell):
    def __init__(self):
        super().__init__()

    @jit
    def construct(self, x, a, i):
        if i > 5:
            return x + a
        else:
            return x

net = Net()
out_me = []
for i in range(1, 10):
    num = mutable(i)
    input_x = ms.Tensor(np.random.rand(2, 3, 4, 5).astype(np.float32))
    input_a = ms.Tensor(np.random.rand(2, 3, 4, 5).astype(np.float32))
    out_me.append(net(input_x, input_a, num))
```

Tuple 或 List 的长度发生改变

当 cell 或者函数的输入有 Tuple 或 List，且长度发生改变时，多次调用该函数或 cell 将会导致多次编译。如以下代码

```
import numpy as np
import mindspore as ms
from mindspore import nn, jit, mutable
```

```
class Net(nn.Cell):
    def __init__(self):
        super().__init__()

    @jit
    def construct(self, x, a, in_me):
        if len(in_me) > 5:
            return x + a
        else:
            return x

net = Net()
in_me = []
out_me = []
for i in range(1, 10):
    in_me.append(i)
    input_x = ms.Tensor(np.random.rand(2, 3, 4, 5).astype(np.float32))
    input_a = ms.Tensor(np.random.rand(2, 3, 4, 5).astype(np.float32))
    out_me.append(net(input_x, input_a, in_me))
```

其中 net 的每次调用之间输入的 list in_me 的长度都不一样，会导致每次调用 net 都重新编图。如以下代码使用 mutable 接口，并将 dynamic_len 参数设置为 True，用 in_me = mutable([], True)代替 in_me = []可以避免多次编图。

```
import numpy as np
import mindspore as ms
from mindspore import nn, jit, mutable

class Net(nn.Cell):
    def __init__(self):
        super().__init__()

    @jit
    def construct(self, x, a, in_me):
        if len(in_me) > 5:
            return x + a
        else:
            return x

net = Net()
in_me = mutable([], True)
out_me = []
for i in range(1, 10):
    in_me.append(i)
    input_x = ms.Tensor(np.random.rand(2, 3, 4, 5).astype(np.float32))
    input_a = ms.Tensor(np.random.rand(2, 3, 4, 5).astype(np.float32))
    out_me.append(net(input_x, input_a, in_me))
```

网络的输入是 tuple[Tensor]、list[Tensor]或 Dict[Tensor]，即使里面 Tensor 的 shape 和 dtype 没有发生变化

当 cell 或者函数的输入有 Tuple 或 List，且元素为 tensor 时，多次调用该函数或 cell 将会导致多次编译。如以下代码

```
import numpy as np
import mindspore as ms
```

```
from mindspore import nn, jit, mutable, Tensor

class Net(nn.Cell):
    def __init__(self):
        super().__init__()

    @jit
    def construct(self, x, a, in_me):
        if len(in_me) > 0:
            return x + a
        else:
            return x

net = Net()
in_me = [Tensor(1), Tensor(2)]
out_me = []
for i in range(1, 10):
    input_x = ms.Tensor(np.random.rand(2, 3, 4, 5).astype(np.float32))
    input_a = ms.Tensor(np.random.rand(2, 3, 4, 5).astype(np.float32))
    out_me.append(net(input_x, input_a, in_me))
```

其中 net 的输入中有 tensor list，会导致每次调用 net 都重新编图。如以下代码使用 mutable 接口，用 in_me = mutable([Tensor(1), Tensor(2)])代替 in_me = [Tensor(1), Tensor(2)]可以避免多次编图。

```
import numpy as np
import mindspore as ms
from mindspore import nn, jit, mutable, Tensor

class Net(nn.Cell):
    def __init__(self):
        super().__init__()

    @jit
    def construct(self, x, a, in_me):
        if len(in_me) > 0:
            return x + a
        else:
            return x

net = Net()
in_me = mutable([Tensor(1), Tensor(2)])
out_me = []
for i in range(1, 10):
    input_x = ms.Tensor(np.random.rand(2, 3, 4, 5).astype(np.float32))
    input_a = ms.Tensor(np.random.rand(2, 3, 4, 5).astype(np.float32))
    out_me.append(net(input_x, input_a, in_me))
```

6.44 编译时报错 ValueError: Please set a unique name for the parameter.

1. 报错信息

编译时报错 ValueError: The value Parameter (name=name_a, shape=(1,), dtype=Float32, requires_grad=True) , its name 'name_a' already exists. Please set a unique name for the parameter.，是什么含义？应该怎么处理？

2. 问题分析

图模式下要求 Parameter 的 name 拥有唯一性，如果存在同名的两个或者多个 Parameter，网络中区分不出不同的对象，将造成错误。我们可以从下面几个角度来排查脚本中的同名的 Parameter，对其中的 Parameter 设置唯一的 name。

```
import mindspore as ms
from mindspore.nn import Cell
from mindspore import Tensor, context, ParameterTuple, Parameter

context.set_context(mode=context.GRAPH_MODE)


class ParamNet(Cell):
    def __init__(self):
        super(ParamNet, self).__init__()
        self.res1 = ParameterTuple((Parameter(Tensor([2], ms.float32),
name="name_a"),
                                    Parameter(Tensor([4], ms.float32), name="name_a")))
        self.param_tuple = (Parameter(Tensor([1], ms.float32), name="name_b"),
                            Parameter(Tensor([2], ms.float32)))
        self.param_list = [Parameter(Tensor([3], ms.float32), name="name_b"),
                           Parameter(Tensor([4], ms.float32))]

    def construct(self):
        out1 = self.res1[0] + self.res1[1]
        out2 = self.param_tuple[0] + self.param_tuple[1] + self.param_list[0] +
self.param_listp[1]
        return out1, out2


net = ParamNet()
res = net()
```

如上脚本，ParameterTuple 中定义了两个同名 name_a 的 Parameter，是不允许的。param_tuple 和 param_list 中定义了同名 name_b 的 Parameter，也是不允许的。还有一种情况是脚本中在同一个 Cell 中实例化某个网络，如下面例子，也将报错 its name 'name_a' already exists. 。

```
import mindspore as ms
import mindspore.nn as nn
from mindspore import Tensor, context, ParameterTuple, Parameter


context.set_context(mode=context.GRAPH_MODE)


class InnerNet(nn.Cell):
    def __init__(self):
        super(InnerNet, self).__init__()
        self.param = Parameter(Tensor([1], ms.float32), name="name_a")
```

```
    def construct(self, x):
        return x + self.param


class OutNet1(nn.Cell):
    def __init__(self, net1, net2):
        super(OutNet1, self).__init__()
        self.param1 = ParameterTuple(net1.get_parameters())
        self.param2 = ParameterTuple(net2.get_parameters())

    def construct(self, x):
        return x + self.param1[0] + self.param2[0]


net1 = InnerNet()
net2 = InnerNet()
out_net = OutNet1(net1, net2)
res = out_net(Tensor([1], ms.float32))
print("res:", res)
```

针对这种情况，我们可以使用 CellList 来管理同一个网络的多个实例。如以下代码：

```
import mindspore as ms
import mindspore.nn as nn
from mindspore import Tensor, context, ParameterTuple, Parameter


context.set_context(mode=context.GRAPH_MODE)


class InnerNet(nn.Cell):
    def __init__(self):
        super(InnerNet, self).__init__()
        self.param = Parameter(Tensor([1], ms.float32), name="name_a")

    def construct(self, x):
        return x + self.param


class OutNet1(nn.Cell):
    def __init__(self, net1, net2):
        super(OutNet1, self).__init__()
        self.cell_list = nn.CellList()
        self.cell_list.append(net1)
        self.cell_list.append(net2)
        self.param1 = ParameterTuple(self.cell_list[0].get_parameters())
        self.param2 = ParameterTuple(self.cell_list[1].get_parameters())

    def construct(self, x):
        return x + self.param1[0] + self.param2[0]


net1 = InnerNet()
net2 = InnerNet()
out_net = OutNet1(net1, net2)
```

```
res = out_net(Tensor([1], ms.float32))
print("res:", res)
```

如下场景，网络 Net 中已经定义了 bias 的 Parameter 和网络实例化再次定义同名的 Parameter，在图模式下是不允许的。

```
import mindspore as ms
ms.set_context(mode=context.GRAPH_MODE)

class Net(ms.nn.Cell):
    def __init__(self, net):
        super().__init__()
        self.net = net
        self.bias = ms.Parameter(ms.Tensor(2.), name='bias')

class SubNet(ms.nn.Cell):
    def __init__(self):
        super().__init__()

sub = SubNet()
net = Net(sub)

net.net.bias = ms.Parameter(ms.Tensor(2.), name='bias')
a = ms.Tensor(3.)
grad_fn = ms.value_and_grad(net, None, net.trainable_params())
print(grad_fn(a))
```

需要为新 parameter 更改名字。

6.45 大模型动态图训练内存优化

1. 内存优化

训练过程中内存开销一般可分为模型参数、优化器状态、前向计算结果和临时变量存储几个维度。内存包括 Host 侧内存及 Device 侧显存，因为显存资源较为稀缺，一般开发者更多关注在显存优化。

2. 用户编程 checklist

• 自定义算子反向数据按需保留：自定义算子易引入内存生命周期过长问题。用户实现自定义算子时，可调用 used_bprop_inputs 仅保存反向需要用到的数据。如下代码所示，其反向仅需要 dout，即 index=2，则需要在__init__函数中添加标识，此时在动态图正向执行完仅会保留 dout 数据。

```
class ReduceScatterToSequenceParallelRegion(nn.Cell):
    "Reduce scatter the input from the model parallel region."

    def __init__(self, need_to_swapaxes):
super(ReduceScatterToSequenceParallelRegion, self).__init__()
        self.world_size = get_tensor_model_parallel_world_size()
self.need_to_swapaxes = need_to_swapaxes
        if self.world_size > 1:
            self.tp_group = get_tensor_model_parallel_group()
self.used_bprop_inputs = [2]#bprop 仅需要保留 dout，则仅将 dout 的 index 传入标记
```

```
    def construct(self, input_):
        if self.world_size == 1:
            return ops.stop_gradient(input_)
        if self.need_to_swapaxes:
            input_ = input_.swapaxes(0, 1)
        output = comm_func.reduce_scatter_tensor(input_, group=self.tp_group)[0]
        if self.need_to_swapaxes:
            output = output.swapaxes(0, 1)
        return output

    # pylint: disable=W0613, C0111
    def bprop(self, *args):
        dout = args[-1]
        if self.world_size == 1:
            return dout
        if self.need_to_swapaxes:
            dout = dout.swapaxes(0, 1)
        output = comm_func.all_gather_into_tensor(dout, group=self.tp_group)[0]
        if self.need_to_swapaxes:
            output = output.swapaxes(0, 1)

        return (output,)
```

• 打开虚拟内存：建议打开虚拟内存，减少大块 Block 的申请导致的显存峰值过高，开启方式如下。值得注意的是，MS_ALLOC_CONF 为 kv 格式，多个配置项需要一次性设置，以逗号分隔，避免多次设置被刷新。

```
export MS_ALLOC_CONF="enable_vmm:True"
```

• 手动开启 GC：建议在每隔几百个 step 时，手动开启垃圾回收机制，它会检测不再使用的对象，并释放它们所占用的内存空间，如下所示。

```
import gc
gc.collect()
```

3. 显存问题分析流程

如上述 checklist 排查和显存配置优化均完成后仍有显存不足的风险，则可根据下列流程分析具体细节。

1. 根据模型计算理论显存值，
2. 打开流同步

```
ms.set_context(pynative_synchronize=True)
```

1. 打开显存抓取配置

```
export MS_ALLOC_CONF="memory_tracker:True"
```

1. 执行获取显存分析报告（二选一）

   • 缩小层数实际运行（推荐）

   • Dryrun 模拟：export MS_SIMULATION_LEVEL=1（注：需要配套 MindSpore 2.5 版本）

根据显存分析报告，重点分析未及时释放的显存。

6.46 大模型精度收敛分析和调优

问题描述

当做完精度对齐后，发现模型精度还是有很大差距或者 loss 不收敛，此时大概率不是框架或者算子的问题，需要针对训练相关的配置做进一步查验。

校验超参

重点检查项：

卡数：对标训练是否使用多卡，使用了几张？

学习率：基础学习率的值，学习率衰减策略，是否使用分组等，一般可以直接打印对比。

batchsize：单卡的 batchsize，多卡的总 batchsize 一般都需要对齐

epoch/step：训练的总 epoch 或者 step

其他检查项：

是否启用 synBN，是否有参数冻结，是否有多个训练阶段，是否在训练过程中更改模型结构等。

此部分注意查看原始论文实验部分，会有训练细节的描述。

校验初始化参数方法

1. 检查每个参数的初始化方法，然后按照[MindSpore 设置初始化参数的方法] 配置初始化方法：

2. 对标脚本中有明确初始化方式的对标实现

3. 对标脚本中无明确初始化方式的查看对标框架 API 找到默认方式对标实现

4. 多卡参数同步

5. 为保持多卡初始化参数一致，需要设置全局 seed 或者设置参数广播：

6. [并行设置参数广播]

7. 预训练模型部分的参数，检查参数是否被覆盖

8. 检查在加载预训练权重后是否有做全局的参数初始化

9. 在训练前打印参数的值，看是否和原始预训练权重的值一致

校验优化器参数

1. 检查基本参数
2. 优化器类型，学习率，weight_decay，momentum 等优化器参数，建议对照优化器 API 逐一查看
3. 检查参数分组策略
4. 对于有有优化器参数分组的情况，检查每个分组的参数是否一致
5. 检查训练参数个数

6. 对于有参数冻结的场景，检查传入优化器的参数是否一致，检查方法

校验混合精度

1. 判断是否使用混合精度

   a. 检查对标脚本是否使用混合精度，torch.cuda.amp

   b. 查看 MindSpore 是否有 float16 的算子，可以在 ir 文件中搜索是否有 float16 执行的算子，一般选取 hwopt_d_end_graph_*.ir

2. 如何添加混合精度

3. 添加混合精度方法

4. 溢出检测方法

5. 当使用混合精度时，算子有可能会有溢出，那如何判断是否会出现溢出呢：溢出检测

当出现溢出时可以怎么做？

1. loss_scale 设置
2. 算子高精度模式:

ms.set_context(ascend_config={“precision_mode”: “force_fp16”)

6.47 Dump 工具应用—算子执行报错，输入数据值越界

1. 问题描述

部分算子在错误的输入下可能会报错，例如 Gather 算子，获取执行下标的数据，如果下标越界，算子访存越界会执行失败。

2. 问题分析

框架默认会开启海思的异常 dump，但其支持的算子有限，执行失败后可以先查看当前目录下是否有 extra-info 目录，如果有数据，即可找海思同事定位。

如果没有数据，需要开启框架的异常 Dump 保存报错算子输入，查看输入是否异常。

3. 配置

配置项：“op_debug_mode”: 4

相关配置：不支持指定 step（iteration 字段不生效），不支持指定算子(kernels 字段不生效)，不支持执行统计信息/Tensor(saved_data 默认为 Tensor），不支持异步。

```
{
    "common_dump_settings": {
        "op_debug_mode": 4,
        "dump_mode": 0,
        "path": "/absolute_path",
        "net_name": "ResNet50",
        "iteration": "all",
        "saved_data": "tensor",
        "input_output": 0,
```

```
        "kernels": ["Default/Conv-op12"],
        "support_device": [0,1,2,3,4,5,6,7]
    },
    "e2e_dump_settings": {
        "enable": true,
        "trans_flag": true
    }
}
```

性能：性能膨胀约 2 倍。

资源占用

磁盘：只存异常算子输出，占用少；

显存：理论上不会增加显存占用；

6.48 Dump 工具应用—网络训练溢出

问题现象

网络训练时可能会发生溢出 loss 或梯度为 nan，需要定位到首个溢出算子。

问题分析

使用溢出 Dump 查看第一个溢出算子。

但有些场景数据是逐步变大，首溢出算子输入很大，这种场景推荐使用统计值 dump 保存全量数据。

配置

配置项：“op_debug_mode”: 3

相关配置：支持指定 step（iteration 字段），不支持指定算子(kernels 字段不生效)，支持统计信息与 Tensor(saved_data 字段）

特定配置：overflow_number，指定溢出 dump 的数据个数。

```
{
    "common_dump_settings": {
        "op_debug_mode": 3,
        "dump_mode": 0,
        "path": "/absolute_path",
        "net_name": "ResNet50",
        "iteration": "all",
        "saved_data": "tensor",
        "input_output": 0,
        "kernels": ["Default/Conv-op12"],
        "support_device": [0,1,2,3,4,5,6,7],
        "statistic_category": ["max", "min", "l2norm"],
        "overflow_number": 0
    },
    "e2e_dump_settings": {
        "trans_flag": true,
```

```
    }
}
```

性能：不溢出时，性能膨胀 2-3 倍，溢出时与溢出数据量有关

资源占用

磁盘

跟溢出数量有关系，可以在 common_dump_settings 配置 overflow_number，指定溢出 dump 的数据个数，执行溢出算子 Dump 的数量。

显存：膨胀不超过 5%。

6.49 模型训练长稳性能抖动或劣化问题经验总结

1. 问题描述

在 VL 类多模态大模型或 CV 类中小模型训练时，由于数据处理逻辑相对复杂，常将 DataLoader 的 num_workers≥4。此时偶尔会遇到模型训练随着时间推移耗时抖动严重，或逐渐劣化，现象参考下图。

2. 原因分析

I/O 瓶颈：受磁盘/云道 OBS 读取速度影响。

CPU 瓶颈：观测 CPU 利用率，如飙升或＞50%，考虑该因素。

锁竞争：考虑是否自定义数据处理方式中存在非进程安全的代码或触发了 GIL 的操作。

内存瓶颈：如 CPU 内存占用率过高，或逐渐升高，考虑是否算子缓存带来的内存占用高间接引入了性能劣化和不稳定现象。

3. 解决经验：

定期垃圾回收：考虑每 200 个 step 手动调用 python 的垃圾回收机制，如 gc.collect()

限制算子缓存大小：当模型为动态 shape 场景，当算子缓存设置过大，易引发内存占用过高的现象，可以考虑设置减小缓存数量（默认 1024），如 export MS_DEV_RUNTIME_CONF=“aclnn_cache_queue_length:128”

绑核。

注：其中周期性耗时长的 step 为手动 GC 的时刻。

6.50 msprobe 工具应用－网络训练溢出

msprobe 工具应用场景－网络训练溢出

1. 问题现象

在动态图场景下，部分算子在错误的输入下可能会溢出。

2. 问题分析

msprobe 进行溢出 dump：使用 msprobe 进行溢出 dump。

3. 配置

1. 用户脚本：

```
from msprobe.mindspore import PrecisionDebugger

debugger = PrecisionDebugger(config_path='./config.json')
# 在训练循环中启动工具
debugger.start()
# 训练循环体
...
# 结束工具
debugger.stop()
```

2.配置项：

MindSpore 场景：

动态图场景：支持 API 级别（"L1"）或 Cell 级别（"L0"）的溢出检测。

静态图场景：支持 Kernel 级别（"L2"）的溢出检测。

MSAdapter 场景：

支持 API 级别（"L1"）或模块级别（"L0"）的溢出检测。

task: 必须设置为 "overflow_check"，表示启用溢出检测功能。

dump_path: 指定溢出数据的存储路径。

rank: 指定需要检测的设备 rank，留空则表示所有 rank。

step: 指定需要检测的训练步骤（iteration），留空则表示所有步骤。

level: 检测的粒度级别：

MSAdapter、MindSpore 动态图: 支持 "L0"（API 级别）或 "L1"（模块级别）。

MindSpore 静态图: 必须设置为 "L2"（kernel 级别），且模型编译优化等级（jit_level）必须为 "O2" 3。

overflow_check: 配置溢出检测的详细参数：

overflow_nums: 最大溢出次数，表示第 N 次溢出后停止检测。默认为 1，设置为 -1 表示持续检测直到训练结束。

check_mode（仅适用于 MindSpore 静态图场景）: 溢出类型，可选 "aicore"、"atomic" 或 "all"，默认为 "all"。

配置示例：

```
{
    "task": "overflow_check",
    "dump_path": "/home/data_dump",
    "rank": [ ],
    "step": [ ],
    "level": "L0",
    "overflow_check": {
```

```
        "overflow_nums": -1
    }
}
```

性能

开启溢出 dump，在不溢出的情况下，性能会膨胀 2-3 倍。如果发生溢出，性能影响则与溢出数据量有关

资源占用

磁盘：只保存溢出算子的输出，占用较少。

显存：理论上不会增加显存占用。

注意事项

1. 路径设置：

   确保 path 字段设置为绝对路径，避免因路径错误导致数据无法保存。

   工具读写的路径（如 config_path、dump_path 等）只能包含大小写字母、数字、下划线、斜杠、点和短横线。

2. 数据采集范围

   确保溢出检测的 API 或模块在 start 和 stop 接口之间被调用，否则数据可能无法被采集。

   如果 API 是通过直接导入（如 from torch import relu）调用的，工具可能无法对其进行 patch，导致数据缺失。

3. 溢出数据的控制

   可以通过配置 overflow_number 控制溢出 dump 的数据个数，当达到指定数值后，溢出数据将不再 dump。

4. 框架与版本限制

   在 MindSpore 框架下，静态图场景的 L2 Level 数据采集需要 MindSpore 版本高于 2.5.0，并且需要使用特定编包选项安装的 whl 包。

5. 数据完整性与格式

   在 dump.json 中，API 或 Module 的统计信息可能出现 null 或 None 值，常见原因包括输入输出为 None、工具不支持某些数据类型，或 Tensor 的 dtype 为 bool。

   在动态图场景下，使用 mindspore.jit 装饰的部分会以静态图模式执行，此时的 Kernel 级数据采集方式与静态图场景相同。

6. 工具性能与兼容性

   使用 msprobe 工具可能会增加运行时的耗时，尤其是在初始化阶段。

   工具在某些场景下可能会出现流同步失败或 GlobalLock 报错，这些问题可能需要修复工具版本或调整配置。

通过以上配置和注意事项，可以有效定位和解决动态图算子溢出问题。

6.51 msprobe 精度定位工具常见问题整理

1 数据采集

dump.json 中 API 或 Module 统计信息里出现 null 或 None 值的原因是什么？

dump.json 里出现 null 或 None 值的可能性较多，常见的场景有：

输入或者输出参数本身是一个 None 值。

输入参数或输出参数类型当前工具不支持，会有日志打印提醒。

输入或者输出 tensor 的 dtype 为 bool 时，Mean 和 Norm 等字段为 null。

如果某个 api 在 dump 支持列表 support_wrap_ops.yaml 中，但没有 dump 该 api 的数据，原因是什么？

首先确认 api 调用是否在采集范围内，即需要在 start 和 stop 接口涵盖的范围内。

其次，由于工具只在被调用时才对 api 进行 patch，从而使得数据可以被 dump 下来。因此当 api 是被直接 import 进行调用时，由于该 api 的地址已经确定，

工具无法再对其进行 patch，故而该 api 数据无法被 dump 下来。

2 精度比对

2.1 常见问题

在同一个目录多次执行 dump 会冲突吗？

答：会，同一个目录多次 dump，会覆盖上一次结果，可以使用 dump_path 参数修改 dump 目录。

如何 dump 算子级的数据？

答：需要配置 level 为 L2 模式。

2.2 异常情况

HCCL 报错： error code: EI0006。

故障现象：使用 msprobe 工具时，报错：error code: EI0006。

故障原因：CANN 软件版本较低导致不兼容。

故障处理：升级新版 CANN 软件版本。

配置 dump_path 后，使用工具报错：[ERROR] The file path /home/xxx/dump contains special characters。

答：请检查你设置的 dump 绝对路径是否包含特殊字符，确保路径名只包含大小写字母、数字、下划线、斜杠、点和短横线；注意，如果执行脚本的路径为 /home/abc++/，设置的 dump_path=“./dump”，工具实际校验的路径为绝对路径 /home/abc++/dump，++ 为特殊字符，会引发本条报错。

无法 dump matmul 权重的反向梯度数据。

答：matmul 期望的输入是二维，当输入不是二维时，会将输入通过 view 操作展成二维，再进行 matmul 运算，因此在反向求导时，backward_hook 能拿到的是 UnsafeViewBackward 这步操作里面数据的梯度信息，取不到 MmBackward 这步操作里面数据的梯度信息，即权重的反向梯度数据。典型的例子有，当 linear 的输入不是

二维，且无 bias 时，会调用 output = input.matmul(weight.t())，因此拿不到 linear 层的 weight 的反向梯度数据。

dump.json 文件中的某些 api 的 dtype 类型为 float16，但是读取此 api 的 npy 文件显示的 dtype 类型为 float32。

答：msprobe 工具在 dump 数据时需要将原始数据从 npu to cpu 上再转换为 numpy 类型，npu to cpu 的逻辑和 gpu to cpu 是保持一致的，都存在 dtype 可能从 float16 变为 float32 类型的情况，如果出现 dtype 不一致的问题，最终采集数据的 dtype 以 pkl 文件为准。

6.52 模型编译的性能优化总结

如果在模型编译过程中发现 step=1 的耗时很长，可分析模型编译是否能进行性能优化。

MindSpore 静态图模式下，部分场景可以通过改变网络写法，使用等价语义替换，或者设置编译选项改变编译机制来优化网络编译性能。

1 规避动态 shape 来优化编译性能

当 step = 1 和 step != 1 的耗时差不多且都很慢时，可能网络在进行多次编图。可能原因为：网络中存在动态 shape。

网络多次编图的判定方法

1.export GLOG_v=1， 开启 mindspore 的 info 日志

2.开始训练，只需要训练几个 step 即可

3.在训练日志中 grep “Start compiling” 和 grep “End compiling”，正常只有一次 Start 和 End。如有多次则网络在多次编图，考虑图中存在动态 shape

动态 shape 的判定方法

当前只能通过打印网络中输入，看是否动态变化来判定网络是否为动态 shape。

之后将推出动态 shape 判定工具。

2 优化编图来优化编译性能

2.1 使用 Select 算子优化编译性能

当图中存在控制流算子时，网络会被切分成多个执行子图，子图间进行流程跳转和数据传递造成模型编译慢。

当分支中算子数量较少时，可通过使用 Select 算子来替换控制流算子，达到优化编译性能。当分支中算子数量较多，建议保留使用 if 语句。

示例代码：

if 语句代码：

```
if ms.ops.reduce_sum(object_masks) == 0:
    stage2_loss = stage2_loss.fill(0.0)
```

修改为使用 select 代码：

```
stage2_loss = ms.ops.select(ms.ops.equal(ms.ops.reduce_sum(object_masks), 0),
stage2_loss.fill(0), stage2_loss,)
```

具体可参考：使用 select 算子优化编译性能

2.2 使用 vmap 优化编译性能

```
mindspore.vmap(` *fn* `, ` *in_axes=0* `, ` *out_axes=0* `)
```

自动向量化（Vectorizing Map，vmap），是一种用于沿参数轴映射函数 fn 的高阶函数。由于自动向量化并不在函数外部执行循环，而是将循环逻辑下沉至函数的各个原语操作中，可获得更好的性能

在处理无依赖关系的批量数据且相关的算子支持 vmap 功能时，可以使用 vmap 替代 for 循环处理批量数据来优化编译性能。

具体可参考：

vmap 详解

vmap 替换 for 循环代码来优化编译性能样例

2.3 使用 HyperMap 优化编译性能

HyperMap 是一个特殊的类，类对象构造时需要传入映射函数 f，调用对象时需要传入 f 的 n 个参数序列，更多使用方法见：HyperMap。映射函数 f 必须是 MultitypeFuncGraph 类型，可参考 MultitypeFuncGraph。

在使用 for 循环批量处理列表元素时，可以通过 HyperMap 等价语义替换来优化网络编译性能。

具体可参考：HyperMap 替换 for 循环代码来优化编译性能样例

3 使用编译缓存优化编译性能

编译缓存的本质是存储了网络模型的编译中间过程文件，当网络模型不变时，生产的编译中间过程文件也是一样的，因此可以复用上一次编程产生的中间过程文件。

mindspore 在模型编译过程中会生成 kernel_meta 等缓存文件。

在进行训练或者推理时，如果某个网络结构未作任何变更，那么可以通过使用编译缓存来缩短编译时间。

方法 1：

脚本中可不删除 kernel_meta 等缓存文件。

方法 2：

设置 set_context(enable_compile_cache=False) 来优化编译性能。

具体可参考：

6.53 大模型动态图训练性能调优指南

大模型具有更多的参数量和更复杂的模型结构，在训练过程中需要耗费更多的计算资源和时间。尤其在 AI 框架动态图模式下易于开发调试带来了极大的灵活性，但也暴露了更多的性能及内存问题。本文将重点介绍昇思动态图下，大模型性能及内存常见的调优方法，以及如何使用工具快速分析并解决瓶颈，帮助大模型开发者们快速上手模型性能调优，提升训练效率，降低训练成本。

1. 典型性能统计指标

1. 单步时间(s)：执行完一个完整 BS 的时间。
2. 吞吐(Samples/s)：网络模型在单位时间内可以处理的最大输入的训练样本数据。throughput =BS *N /step _time。其中，BS 为每个数据并行维度的 batch size 大小，N 为集群中数据并行维度的大小，step_time 为在分布式集群中，执行完一个完整 BS 的时间（单位为 s）。
3. MFU(%)：Model FLOPs Utilization，即模型算力利用率，是指模型一次前反向计算消耗的矩阵算力与机器算力的比值。它直接反映了模型在训练过程中对计算资源的有效利用程度。MFU 的值受到多种因素的影响，包括但不限于：

   模型架构：不同架构的模型在分布式训练中的算力利用率可能存在显著差异。例如，Transformer 架构的模型由于其并行性和矩阵运算密集的特点，通常能够在分布式训练中实现较高的 MFU。

   分布式训练策略：包括数据并行、模型并行、流水线并行等不同的并行策略及这些策略的具体实现方式（如梯度累积、张量并行等），都会对 MFU 产生重要影响。

   硬件环境：显卡型号、数量、网络连接速度等硬件因素都会限制分布式训练的性能和算力利用率。

   软件优化：包括编译器优化、库函数优化、自动混合精度训练等软件层面的优化措施，也能在一定程度上提升 MFU。

   数据集和批次大小：数据集的大小和复杂性，以及训练时使用的批次大小，也均会影响每次迭代中的计算量和算力利用率。
4. 线性度：单卡训练扩展到多卡、多机多集群后的效率度量指标称为线性度，又名加速比。一般根据吞吐率计算得到。即集群线性度为多机总吞吐率/（单机吞吐率*集群卡数）。线性度的取值范围为 0~1，数值越接近于 1，其性能指标越好，一般大于 0.8 认为较优。

2. MindSpore 动态图机制介绍

动态图 PyNative 框架

MindSpore PyNative 动态图模式采用 pybind 算子直调方法，提升 API 性能。2.3 版本前，大部分 API 使用了小算子进行拼接，同时采用了单算子子图进行执行，需要进行单算子子图构图、编译优化、算子选择、算子编译等一系列操作，首轮性能较差。针对这两点进行性能优化，提出了算子直调的方式，即正向算子执行直接 pybind 调用到底层算子接口，减少整体流程和数据结构转换开销。不同 API，性能提升 0.5~4 倍。此外，提供了基础分布式能力接口，如硬件相关接口、重计算接口、通信基础接口等。

详细介绍见：动态图支持算子直调

3. 性能调优

为了优化前文介绍的性能度量指标，提升训练效率，一般从如下几个维度拆解优化。

• 数据处理耗时：指模型加载训练数据和权重的时间，包括将数据从硬件存储设备读取到 CPU、在 CPU 中进行数据的预处理、以及 CPU 数据传输到 NPU 的过程。对于需要切分到若干张 NPU 上的模型，数据加载时间还包括从一张 NPU 广播到其他 NPU 上的时间。

• Host 下发耗时：指 Python 侧脚本逻辑及 api 接口 launch 的时间，一般希望这部分耗时与 Device 执行流水起来，避免 Device 空闲。

• Device 执行耗时：NPU 侧执行各个算子计算逻辑的时间。

• 通信耗时：指在分布式训练中，设备之间进行通信传输的时间，可以通过优化网络拓扑、减少通信频率等方式来减少通信耗时。同时，通信和计算通常可以并行执行，并行库提供的并行技术会保证部分通信时间被掩盖。

4. 用户编程 checklist

• 关闭确定性计算

ms.set_context(deterministic='OFF')

• 关闭流同步

ms.set_context(pynative_synchronize=False)

• 使用高性能 API

• mint：MindSpore 提供了对标 PyTorch 的 mint 系列接口，绝大多数情况下，其性能会持平或高于原 ops 系列接口。详情参考 API 列表：mint 接口列表

• 大模型融合算子，如 RotaryPositionEmbedding、Swiglu 等。

• 避免冗余的数据拷贝

• 尽量采用原地更新接口，减少冗余 Tensormove 操作

• 减少不必要的转连续操作：当前 aclnn 算子大多支持非连续输入，尽量减少在脚本中大量使用.contiguous()，或可以先 is_contiguous()判断后再调用。

• 避免频繁数据拷贝：需要数据拷贝时，尽量采用.from_numpy 接口，当数据连续时，会通过免拷贝方式将 Numpy 数组转换为张量。mindspore.Tensor.from_numpy

• 减少原生 Python 累加等函数使用，如.sum()（建议替换成 mint.sum()）

5. 高阶优化特性

• JIT：将 Python 函数编译为可调用的 MindSpore 静态图，MindSpore 可以在运行时对图进行优化，提升该模块性能，一般选择在优化器构造函数处添加@jit 标签。详见：jit 介绍

6. 性能分析案例

MindStudio Insight 用户指南

7 模型切分问题案例

7.1 MindSpore 权重自动切分后报错 ValueError: Failed to read the checkpoint. please check the correct of the file.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.0

执行模式： 不限

2. 报错信息

2.1 问题描述

原始 ckpt 大小 59G，按照 model parallel=2, pipeline parallel=4 的设置，将模型自动切分后，发现切分后的 8 个模型大小均在 50G 左右，导致加载模型的时候报 MemeryError。

2.2 报错信息

```
ValueError: Failed to read the checkpoint file /mnt/checkpoint_7.ckpt. May not have permission to read it, please check the correct of the file.
```

3. 根因分析

分析原因在于，yaml 文件中将 layer 设置为了 4，但是原始模型文件 layer 有 40 层，导致生成的切分策略文件只涉及前 4 层，所以在自动切分的时候，只能切分前 4 层，而后面的 36 层均附在了后面，导致了切分后的每个文件都很大。

4. 解决方案

将 layer 设置为 40 层进行切分，或者如果模型太大，可以先手工降低原始模型层数，然后再进行切分。

7.2 MindSpore2 机自动切分模型权重报错 NotADirectoryError: ./output/transformed chec kpoint/rank_15 is not a real directory

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.0

执行模式： 不限

2. 报错信息

2.1 问题描述

在 2 机配置权重自动切分时候遇到如下报错。

2.2 报错信息

```
2023-09-22 01:57:56,963 - mindformers[utils.py:387] - TNFO -........Collecting
strategy........
2023-09-22 01:57:56,970 - mindformers[utils.py:411] - WARNING - Can't collecting
all strategy, device num > 8!
Rank 15: Waiting trans forming ckpt........
Rank 15: Transform succeed!
2023-09-22 03:57:57,751 - mindformers[utils.py:534] - INFO - ........Start load
checkpoint from checkpoint........
2023-09-22 03:57:57,787 - mindformers[utils.py:242] - INFO - When distributed loads
aresliced weights,load_ checkpoint should be a checkpoint directory containing thry
of rank_{0-*},The directory structure is as follows: **chec
kpoint_root_dir/checkpoint/ rank_{0-*}/**.ckpt
[WARNING] MD (75354, fffd2e7 fcle0,python) :2023-09-22-03:57:58.883.267
[mindspore/ccs rc/minddata/dataset/engine/perf/auto_tune.cc: 152]
SaveAutotuneConfig] File: <./autdtune_15.json> already exists. File will be
overwritten with the AutoTuned data pipeline configuration.
[WARNING] MD (75354, ffffaaf35c40,python) :2023-09-22-03:57 :58.885.382
[mindspore/ccsrc/minddata/dataset/engine/dat asetops/data queue op.cc: 115]
~DataQueue0p] preprocess_batch: 100;
|batch_queue: 1, 1, 1, 1, 1, 1, 1, 1, 1, 1;
                  push_start_time -> push_end_ time
Traceback (most recent call last):
   File "wizardcoder/ run_wizardcoder.py", line 148, in <module>
       device_id=args.device_id)
```

```
    File "wizardcoder/run_wiza rdcoder.py", line 90, in main
        task.finetune(finetune_checkpoin t=config.load_checkpoint, auto_t
rans_ckpt=config.auto_trans_ckpt, resume=resume)
    File "/home/wizardcoder/1_wizardcoder-mindformers-916/mindformers/t
rainer/trainer.py", l ine 522, in finetune
        is_full_config=True, **kwargs)
    File "/home/wizardcoder/1_wizardcoder-mindformers -
916/mindformers/trainer/causal_language_model ing/causal_language_modeling.py",
line 106, in train
        **kwargs)
    File "/home/wizardcoder/1_wizardcoder-mindforme rs-
916/mindformers/trainer/base_t rainer.py", line 616, in training_process
        transform_and_load_checkpoint (config, model, network, dataset)
    File "/home/wizardcoder/1_wizardcoder-mindformers-916/mindformers/t
rainer/utils.py", line 309, in transform_and_load_checkpoint
        load_ckpt(config, network, optimizer=optimizer)
    File "/home/wizardcoder/1_ wizardcoder-mindformers-
916/mindformers/trainer/utils .py", line 536, in load ckpt
        checkpoint dict = load distributed checkpoint (config)
    File "/home/wizardcoder/1_wlzardcoder-mindformers-
916/mindformers/trainer/utils.py", line 247, in load_dis tributed_checkpoint
        distribute checkpoint path = get_last_checkpoint(distribute_ checkpoint_dir)
    File "/home/wizardcoder/1_wizardcoder-mindformers -
916/mindformers/trainer/utils.py", line 552, in get_last_checkpoint
        f"{chockpoint_dir} is not a real directory,
NotADirectoryError: ./output/transformed chec kpoint/rank_15 is not a real
directory,When distributed loads are sliced weights , load_checkpoint should be a
checkpoint containing the directory of rank {0-*},The directory structure is as
follows: **checkpoint root dir/rank {0-* }/checkpoint/**.ckpt
```

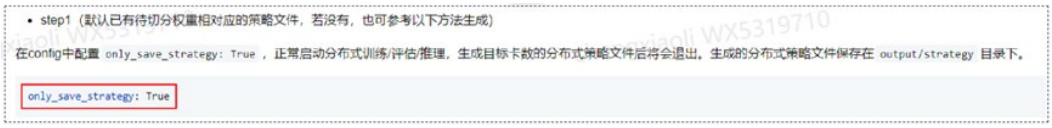
3. 根因分析

因为每台机器只会生成自己的 strategy 文件，而看不到其它机器的 strategy 文件，导致后续自动切分时缺少全部的 strategy 文件而切分失败。

4. 解决方案

因此在多机并行时，只能采用离线切分的方法。具体操作如下：

1. 台机器各自生成自身的 strategy 文件，只需要配置 only_save_strategy 为 True 即可，此时 auto_trans_ckpt 不起作用，不会自动切分权重。

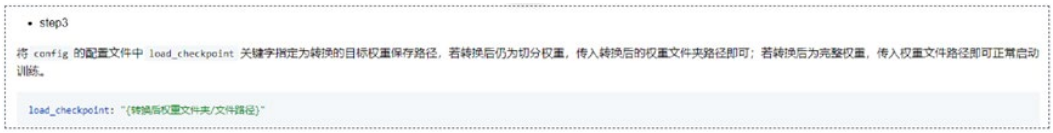


2. 将 2 台机器的 strategy 文件统一汇总到一台机器的一个文件夹下（假设文件夹名称为/home/strategy/），再在这台机器执行图中的权重切分脚本。

注意：初始权重是完整网络权重，所以不需要写入--src_ckpt_strategy，所以正确的脚本启动语句如下：（--src_ckpt_dir 是完整脚本的路径，里面还有一层 rank_0 文件夹，其中存放完整 ckpt 文件；--dst_ckpt_dir 存放切分后的模型）

```
python mindformers/tools/transform_ckpt.py --dst_ckpt_strategy /home/strategy/ --
src_ckpt_dir
/home/mindspore_models/ --dst_ckpt_dir /home/distribute_models/
```

3. 切好模型后，将模型传给另一台机器中，保证 2 台机器都有切分后的模型；之后，配置 yaml 文件的 load_checkpoint 为上述 dst_ckpt_dir 文件夹位置，此时 auto_trans_ckpt 一直保持为 False。



7.3 Mindspore 并行策略下 hccl_tools 工具使用报错

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.1

执行模式： 不限

2. 报错信息

2.1 问题描述

并行策略下使用 hccl_tools 工具，报错信息如下。

2.2 报错信息

```
[root@localhos t 1_wizardcoder-mindformersl# cd research/
[root@localhost researchl# python ../mindformers/tools/hccl_tools.py --device_num
"[0,8)"
start ../mindformers/tools/hccl_tools.py
visible_devices:['0', '1', '2',T13', '4, 15', 16', 17,]
server_id:127.0.0.1
device num_list: [0, 1, 2, 3, 4, 5, 6, 7]
Command execute failed!
Failed to call hccn_tool, try to read /etc/hccn.conf instead
Traceback (most recent cal l last):
   File "../mindformers/tools/hccl_tools.py", line 167, in <module>
      main()
   File "../mindformers/tools /hccl_tools.py", line 137, in main
      device_ip = device_ipsldevice idl
KeyError: '0'
```

3. 根因分析

这是因为/etc/hccn.conf 文件不存在，需要创建此文件。

4. 解决方案

```
/etc/hccn.conf 文件不存在，创建此文件。
[root@localhost researchl# vi /etc/hccn.conf
[root@localhost research]# hccn_tool -i 0 -ip -s address 90.90.92. 101 netmask
255.255.255.0
[root@localhost researchl# cat letc/hccn.conf

address_ 0=90.90.92.101
```

```
netmask 0=255.255.255.0
[root@localhost research]# hccn_tool -i 1 -ip -s address 90.90.93.101 netmask
255.255.255.0
[root@localhost research]# hccn_tool -i 2 -ip -s address 90.90.94.101 netmask
255.255.255.0
[root@localhost research]# hccn_tool -i 3 -ip -s address 90.90.94.101 netmask
255.255.255.0
[root@localhost research]# hccn_tool -i 4 -ip -s address 90.90.94.101 netmask
255.255.255.0
[root@localhost research]# hccn_tool -i 5 -ip -s address 90.90.94.101 netmask
255.255.255.0
[root@localhost research]# hccn_tool -i 6 -ip -s address 90.90.94.101 netmask
255.255.255.0
[root@localhost research]# hccn_tool -i 7 -ip -s address 90.90.94.101 netmask
255.255.255.0
[root@localhost research]# cat /etc/hccn.conf
```

7.4 MindSpore 大模型报错: Inner Error! EZ9999 [InferShape] The k-axis of a(131072) and b(16384) tensors must be the same.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.0

执行模式：Graph

2. 问题描述

```
Ascend Error Message:
Ez9999 Inner Error!
EZ9999 [InferShape] The k-axisof a(131072) and b(16384) tensors must be the
same[FUNC: InferShape] [FILE:matrix calculation ops.cc][LINE:934]
       TraceBack (most recent call last):
       Failed to infer output shape [FIINC:MatMuly2InferShapel
[FTIE:matrix_calculation_ops_ccl [LTNE:2253]
       Call InfershapeAndType for node:Default/backbone-
CausalLMHydrawithValueHead/v head2 -Dense/MatMul- op5643 (MatMulV2)
    failed[FUNC:Infer][FILE:infershape_pass.cc][LINE:119]
       process pass InterShapePasson node:Detault/backbone-
CausaLLMHydraWithValueHead/v_head2 -Dense/MatMul-op5643 tailed,
ret:4294967295[FUNC:RunPassesOnNode][FILE:base_pass -cc] [LINE:571]
       [Call] [PreRun] Failed, graph_id:7, session_id:0.
[FUNC :CompileGraph][FILE:graph_manager .cc][LINE:4111
       [Compile] [GraphlCompile graph failed, error code:1343225857, session_id:0,
graph_id:7. [FUNC:CompileGraph] [FILE:ge_api.cc][LINE:1150]
(Please search "Ascend Error Message" at https://www.nlindspore.cn for error code
description)
```

3. 报错信息

报错信息为 infershape 失败，shape 通过计算发现差 8 倍，遇到 infershape 失败这种问题，首先怀疑是切分策略哪里不对导致。

dp=1，mp=8，怀疑是切分策略导致，保存 ir 图。

```
context.set_context(save_graphs=3, save_graphs_path='./graph')
```

查看 ir 图，找到对应报错算子的切分策略，发现确实是按 8 进行切分。

查看代码，发现代码并没指定切分策略，按逻辑应该是按 dp 1 进行切分。

原因：v_dense2 的输入来自于 mul，mul 是指定切分策略，执行序导致 v_head 的切分策略也变成按 mp8 进行切分，导致报错。

4. 解决方案

显式的配置切分策略，代码修改为：

```
self.v_head0 = Linear(model_config.hidden_size, 2 * model_config.hidden_size,
weight_init=TruncatedNormal(0.02), activation="relu", has bias=True).shard(((1, 1),
(1, 1)))
self.v_head2 = Linear(2*model_ config.hidden_size, 1,
weight_init=TruncatedNormal(0.02), has_bias=True).shard(((1, 1), (1, 1)))
```

7.5 Transformers 报错 google.protobuf.message.DecodeError: Wrong wire type in tag.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.10

执行模式： 不限

Python 版本: 3.8.15

操作系统平台: linux

2. 报错信息

Transformers 模型 finetune 加载权重，自动切分报错。

报错日志：

```
google.protobuf.message.DecodeError: Wrong wire type in tag.
```

3. 根因分析

之前下载的权重文件损坏 or 下载的权重文件不完整，导致解析失败，并非文件权限问题。

4. 解决方案

删除权重后，重新拉取权重后切分成功。

8 其他类型问题案例

8.1 报错日志不完整

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 问题描述

报错日志不完整

3. 报错信息

通常我们需要通过打开 mindspore 和海思的日志来定位更为详细的日志，打开方法如下：

```
export GLOG_v=1 -->查看 mindspore 日志 0debug, 1info, 2warning,3error
export ASCEND_GLOBAL_LOG_LEVEL=1 --> 查看 cann 日志
export ASCEND_SLOG_PRINT_TO_STDOUT=0 --> 0 不输出， 1 打屏
```

定位问题时，主要需要开启以下环境配置参数：

```
export ASCEND_GLOBAL_LOG_LEVEL=3 □  0-debug、1-info、2-warning、3-error
export ASCEND_SLOG_PRINT_TO_STDOUT=1 □  0 不输出， 1 打屏
export ASCEND_GLOBAL_EVENT_ENABLE=0 □  设为 0 后不打印 EVENT
```

8.2 日志显示没有成功加载预训练模型：model built, but weights is unloaded, since the config has no attribute or is None.

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 问题描述

日志显示没有成功加载预训练模型

```
model built, but weights is unloaded, since the config has no attribute or is None.
```

3. 解决方案

模型 mode 如果为 train 或者 finetune，则该阶段其实不需要设置 checkpoint_name_or_path，所以这里的日志正常，只要在 load_checkpoint 设置了模型地址即可。

8.3 工作目录问题：'from mindformers import Trainer'报错 ModuleNotFoundError:No module named ' mindformers'

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 报错信息

2.1 问题描述

将待迁移模型 wizardcoder 放到 mindformers/research 文件夹下，运行 run_wizardcoder.py 文件时，报错“no module named ‘mindformers’”

2.2 报错信息

```
Traceback (most recent call last):
  File "run_wizardcoder .py", line 20, in <module>
```

```
   from mindformers import Trainer
ModuleNotFoundError:No module named ' mindformers'
```

3. 根因分析

这种情况是因为工作目录问题，因为运行代码在 research 文件夹下，找不到 mindformers 模块在哪个路径。

4. 解决方案

需要在 run_wizardcoder.py 代码开头添加路径引用语句。

```
import os
import sys
sys.path.append(os.path.abspath("../.. "))
```

需要注意第 3 行，因为 mindformers 文件夹在当前 run_wizardcoder.py 文件的外两层处，所以需要使用"../.."来表示这个路径。

8.4 NFS 上生成 mindrecord 报错 Failed to write mindrecord meta files

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2

执行模式： 不限

2. 报错信息

模型正向报错信息如下：

```
start commit !!
[ERROR] MD(68811,7f2 382c85700,python):2023-11-27- 16:03:22.406.216
[mindspore/ccsrc/minddata/mindrecord/ io/shard_index_generator.cc:587]
DatabaseWriter] Exception thrown from dataset pipeline. Refer to 'Dataset Pipeline
Error Message'.    [Internal ERROR] Failed to execute the sql [ CREATE TABLE
INDEXES( ROW ID   INT NOT NULL, PAGE ID RAW  INT NOT NULL, PAGE OFFSET RAW  INT NOT
NULL, PAGE_OFFSET_RAW_END  INT NOT NULL, ROW_GROUP_ID INT NOT NULL, PAGE_ID_BLOB INT
NOT NULL, PAGE_OFFSET_BLOB INT NOT NULL, PAGE_OFFSET_BLOB_END INT   NOT NULL,-PRIMARY
KEY(ROW_ID)); ], database is locked
Line of code : 131
File: mindspore/ccsrc/minddata/mindrecord/io/shardindex_generator.cc
Traceback (most recent call last):
   File "preprocess_data.py", line252, in <module>
      main()
   File "preprocess_data  py", line 221, in main
      writer.commit()
   File "/root/miniconda3/envs/lc j_new/lib/python3.7/s ite-
packages/mindspore/mindrecord/filewriter.py" , line 455, in commit
      self._generator.wr ite_to db()
   File "/root/miniconda3/envs/lcj_new/l ib/python3.7/s ite-
packages/mindspore/mindrecord/shard indexgenerator.py", line 70, in write_to_db
```

```
       ret = self._generator.write_to_db()
RuntimeError: Exception thrown fromdataset pipeline. Refer to 'Dataset Pipeline
Error Message'.
- Framework Unexpected Exception Raised:
This exception is caused by framework's unexpected error. Please create an issue at
https://gitee.com/mindspore/mindspore/issues to get help.
- Dataset Pipeline Error Message:
[ERROR] [Internal ERROR] Failed to write mindrecord meta files.
- C++ Call Stack: (For framework developers)
mindspore/ccsrc/minddata/mindrecord/ io/shard index_generator.cc(576).
```

3. 根因分析

Sqlite 暂不支持对 NFS 上的网络文件进行操作，请用户使用本地磁盘进行 mindrecord 读写。

```
6.0 How To Corrupt Your Database Files
The pager module is very robust butit can be subverted. This section attempts to
identify and explain the risks.(See also the Things That Can GoWrong section of the
article on Atomic Commit.
Clearly, a hardware or operatingsystem fault that introduces incorrect data into
the middle of thedatabase file or journal will causeproblems. Likewise, if a
rogueprocess opens a database file or journal and writes malformed data into the
middle of it, then the database will become corrupt.There is not much that can
bedone about these kinds of problemsso they are given no further attention.
SQLite uses POSIX advisory locksto implement locking on Unix. OnWindows it uses the
LockFile(),LockFileEx(), and UnlockFile() system calls. SQLite assumes that these
system calls all work as advertised. If that is not the case, then database
corruption can resuIt. One should note that POSIX advisory locking is known to be
buggy or even unimplemented on many NFS implementations (includingrecent versions
of Mac OS X) andthat there are reports of locking problems for network filesıvstems
under Windows. Your best defense is to not use SQLitefor files on a network
filesystem.
```

4. 解决方案

mindrecord 存储的路径，避免在 NFS 上，应该指向本地磁盘。

8.5 盘古-智子 38B 在昇腾 910 上，greedy 模式下无法固定输出

1. 报错信息

- 使用盘古智子 38B V18 版本在昇腾 910 上推理，结果有随机性，无法实现固定输出（greedy 模式）。
- 想要得到 greedy 模式下每次输出一致。

3. 根因分析

- greedy 模式主要是将后处理的下面 3 个参数修改成如下值：

```
TOP_K_NUM=1
TOP_P=0.0 # greedy: 0.0; sampling: 1.0
TEMPERATURE=0.3
```

- 当前端到端输出不一致，因为包含前后处理，所以先在 model.predict 接口前后打点，确认 predict 接口的输入和输出在多次调用下值是否一致。
- 根据保存的输入输出数据，发现输入数据一致，但是输出数据(logits)不一致。
- 考虑到端到端的回答内容虽然存在不一致，但是内容上都属于合理的回答，所以应该不属于精度问题，而是推理过程中某些随机性导致的计算结果的随机性，进而导致最终答案的随机性。
- 推理过程中可能涉及到随机性的有：
  a. 原理上有随机性的算子（如 Dropout）
  b. 多卡通信涉及到的算子（如 AllReduce）
  c. 计算上存在随机性的算子（atomic 类算子，如 ReduceSum, Matmul 等）
- 由于是推理过程，而不是训练，所以模型中应该不存在 Dropout 等原理上有随机性的算子。排查后排除了上面第一条的影响。
- 通过在启动脚本中加入如下环境变量，可以排除上面第二条的影响：

```
export HCCL_DETERMINISTIC=true
```

- 通过在 config.ini 中增加 ge.deterministic，用来开启算子的计算确定性。
- 增加上述配置以后，发现算子的计算确定性没有开启成功。

3. 解决方案

是当前 CANN 版本 MatMul 算子确定性有问题，建议升级 CANN 版本。升级到 MindSpore 2.2 + CANN7.0 + 对应的智子版本后，gready 模式下，输出一致。

8.6 昇腾上 moxing 拷贝超时导致 Notify 超时

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend910

软件环境: MindSpore 版本: 2.2.0

执行模式： Graph

Python 版本: 3.7.6

操作系统平台: linux

2. 报错信息

```
- Ascend Error Message:
EE9999: Inner Error!
EE9999 The error from device(chipld:0,dield:0), serial number is 1, notify wait
timeout occurredduring task execution, stream id:28, sq id:28,task_id:2,
notify_id=5, timeout=1836.[FUNC:ProcessStarsWaitTimeoutErrorlnfo][FILE:device_error
_ proc.cc][LINE:1308]
    TraceBack (most recent call last)
    Notify wait execute failed, device id=0, stream_ id=28,task_id=2, fip_ num=0,
notify_id=5[FUNC:GetEnor][FILE:stream.cc][LINE:1483]
    rtStreamSynchronizeWithTimeout execute failed,reason=[the model stream
executefailed][FUNC:FuncErrorReason][FILE:error_message_manage.cc][LINE:50]
```

```
    synchronize stream failed, runtime result = 507011[FUNC
ReporCallEror][FILE:log_ inner.cpp][LINE:161)

(Please search "Ascend Error Message" athttps:/www.mindspore.cn for error code
description)
- C++ Call Stack. (For framework developers)
mindspore/ccsrc/runtime/graph schedulerlgraphscheduler.cc:679 Run
```

3. 解决方案

- 中间过程 notify 超时。查看日志，发现报错是在 start upload output file to obs 后，查看 mindformers 代码后，该日志是通过 moxing 上传 ckpt 等输出文件到 obs，定位后发现上传 obs 时间过长，超过 HCCL 等待时间，阻塞了网络训练导致 notify 等待超时

解决方案：

1、推动云道定位 moxing 上传过慢问题

规避方式: 不手动调用 moxing 上传，通过修改 ckpt 保存路径到云道支持的 output 路径，可以实现自动上传

step1：修改 mindformers config 文件，删除 ObsMonitor

```
# callbacks
callbacks:
    - type: MFLossMonitor
    - type: CheckpointMointor
      prefix: "1lama 70b"
      save_checkpoint_steps: 1000
      integrated_save: False
      async_save: False
    - type: ObsMonitor
```

step2: 修改默认 ckpt 保存路径：mindformers/tools/utils.py, 修改 MA_OUTPUT_PATH 为: /home/ma-user/modelarts/outputs/train_url_0

```
MA_OUTPUT_ROOT = '/cache/ma-user-work'
```

使用的前提是，拉起任务时有设置 output 路径。设置后，训练生成的文件会被自动上传到输出数据的 obs 路径，上传文件和模型训练过程解耦。

8.7 MTP Ascend910 切换不同型号设备报错：KeyError：‘group_list’

1. 系统环境

硬件环境(Ascend/GPU/CPU): Ascend

软件环境: MindSpore 版本: 2.2.10

执行模式： 不限

Python 版本: 3.8.15

操作系统平台: linux

2. 报错信息

```
Traceback (most recent call last):
   File "/opt/huawei/schedule-train/algorithm/run_ascend910/run_ascend910.py", line
24, in <module>
       rank_table = RankTableTemplate1(rank_table_path_origin)
   File "/opt/huawei/schedule-train/algorithm/run_ascend910/rank_table.py" , line
123, in _init
       json_data["group_list"][0]["instance list"l = sorted(json data["group
list"][Ø]["instance list"],
KeyError: 'group list'
```

3. 根因分析

不同型号设备的获取 rank_table 方式不同。

4. 解决方案

如果采用如下获取 rank_table 方法报错：

```
rank_table = RankTableTemplate1(rank_table_path_origin)
```

那么需要修改为：

```
from rank_table import RankTable, RankTableTemplate2
rank_table = RankTableTemplate2(rank_table_path_origin)
```