MindSpore

ME-Compiler SIG Meeting

作者：ME-Compiler SIG成员

目录

contents

MindSpore

嘉宾介绍

MindSpore

Yi Yang：深圳湾实验室副研究员，华为首批HAE(Huawei Ascend Expert)开发者。

PengXiang Xu：鹏城实验室副研究员， MindSpore社区活跃开发者。

Jenks Huang：ME-Compiler开发工程师，负责图优化开发。

KinFung Yu：ME-Compiler开发工程师，负责自动微分开发。

Felix Chen：ME-Compiler开发工程师，负责控制流语法表达开发。

MindSpore默认执行模式变更

MindSpore

执行模式管理

MindSpore支持PyNative和Graph这两种运行模式：

- `PYNATIVE_MODE`：动态图模式，将神经网络中的各个算子逐一下发执行，方便用户编写和调试神经网络模型。
- `GRAPH_MODE`：静态图模式或者图模式，将神经网络模型编译成一整张图，然后下发执行。该模式利用图优化等技术提高运行性能，同时有助于规模部署和跨平台运行。

Q:PyNative模式和Graph模式的区别?

A: 在使用效率上，两个模式使用的算子是一致的，因此相同的网络和算子，分别在两个模式下执行时，精度效果是一致的。由于执行机理的差异，网络的执行性能是会不同的，并且在理论上，MindSpore提供的算子同时支持PyNative模式和Graph模式；

在场景使用方面，Graph模式需要一开始就构建好网络结构，然后框架做整图优化和执行，对于网络固定没有变化，且需要高性能的场景比较适合；

在不同硬件（`Ascend`、`GPU`和`CPU`）资源上都支持这两种模式；

代码调试方面，由于是逐行执行算子，因此用户可以直接调试Python代码，在代码中任意位置打断点查看对应算子`/api`的输出或执行结果。而Graph模式由于在构造函数里只是完成网络构造，实际没有执行，因此在`construct`函数里打断点是无法获取对应算子的输出，而只能等整网执行中指定对应算子的输出打印，在网络执行完成后进行查看。

资料链接：https://www.mindspore.cn/doc/programming_guide/zh-CN/master/context.html
https://www.mindspore.cn/doc/faq/zh-CN/master/backend_compile.html

MindSpore默认执行模式变更

MindSpore

默认执行模式修改为Graph模式

1. Mindspore框架主模式为Graph模式，但当前默认执行模式为PyNative模式。

2. Graph模式和PyNative模式在某些场景下无法直接无缝切换，在语法层面，Graph模式比PyNative模式“更严格”。用户在使用PyNative模式完成代码调试后，直接切换Graph模式，会高概率出现失败情况。

3. 默认使用Graph模式，需要调试等用途时修改为PyNative模式，是更加合适的模式切换方式。

自动微分现状与规划

MindSpore

当前MindSpore采用Reverse自动微分机制，支持：

- 调用GradOperation求函数对输入的导数
- 反复调用GradOperation求函数对输入的高阶导数

还支持一些使用上的小技巧：
- 传入sens值对网络输出值做缩放以修改梯度
- 使用stop_gradient做反向传播的剪枝
- 自定义Cell的梯度函数

后续规划特性：

- 提供更多高阶接口，方便用户使用
- 考虑使用高性能高阶微分方案，提升求高阶导数的效率
- 提供Forward模式和Forward- Reverse混合模式的自动微分机制，提升求导效率

疑问与建议收集

MindSpore

THANK YOU!