**题目： 基于pytorch的任意精度模拟训练框架研究**

**创新点：**

-提出了一种可分组，支持任意精度位宽模拟的GEMM算子，内部计算格式可采用纯FP32或纯FP16，位宽低于FP16的精度格式采用FP16格式模拟。

-在pytorch中做了算子嵌入，打通了推理和训练流程，使模型推理和训练可以直接使用新的GEMM算子。

-基于arr论文，实现了推理或训练过程中各项数据的统计，如swamp数，各层均值/方差/数据分布。通过提供充分的统计数据，来支持用户探索模型精度的下限。

**背景和动机：**

-GEMM算子是AI模型的核心计算，通常GEMM计算占比高达95%以上，降低精度可以有效减少GEMM的计算时间，同时减少模型的内存占用，进而加速模型推理和计算，降低成本。但降低精度会对计算准确度有所损害，精度过低会导致模型完全不可用，由于缺少可靠的模拟。当前的一些模型对降低精度选择过于保守的策略，如FP16-FP32混合精度。

-使用构建特定硬件的方式模拟将会造成大量开销，为了避免这些开销，我们需要构建软件模拟 。

-目前相关的工作普遍采用精度截断的方式来评估降低精度造成的影响，真正计算时还是采用FP32或混合精度。包括Nvidia的apex库，号称纯FP16的矩阵乘算子，内部计算部分和累加时格式仍为FP32，这使得模拟的情况过于理想，降低精度的影响无法得到有效的评估。

-此外，目前的模拟工作大多基于经验，需要不断的调整精度格式来探索精度下限。

-基于这种背景，我们提出了一种可分组，支持任意精度位宽模拟的GEMM，内部计算可采用纯FP32/混合精度/纯FP16（包括部分和累加）。第一个挑战是，在保证正确性和计算模式的多样性时，还要保证计算的速度。另一个挑战是，我们需要最大限度地减少将现有代码转换到研究低精度训练代码所需的努力 。第三个挑战是为了减少不断降低精度带来的时间开销，我们结合理论，实现了推理和训练过程中各项数据的分层统计，如swamp数，各层均值/方差/数据分布。

**理论部分：**

* 参照vrr论文，提出自己的公式。重点在于经验公式v(n)的修改。

**模块设计：**

1.GEMM模块：

（分点介绍算子如何实现分组加/纯FP16的部分和累加/低精度模拟）

1. 分组加：
2. 内部低精度实现
3. 更低精度的模拟实现:

2.算子模块：

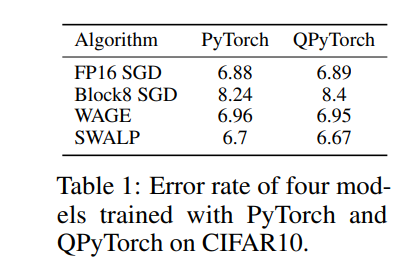
介绍如何基于GEMM，实现conv/linear等算子并实现推理/训练。

3．其他模块:

各项数据的分层统计(swamp/均值/方差/数据分布/vrr/v(n) )

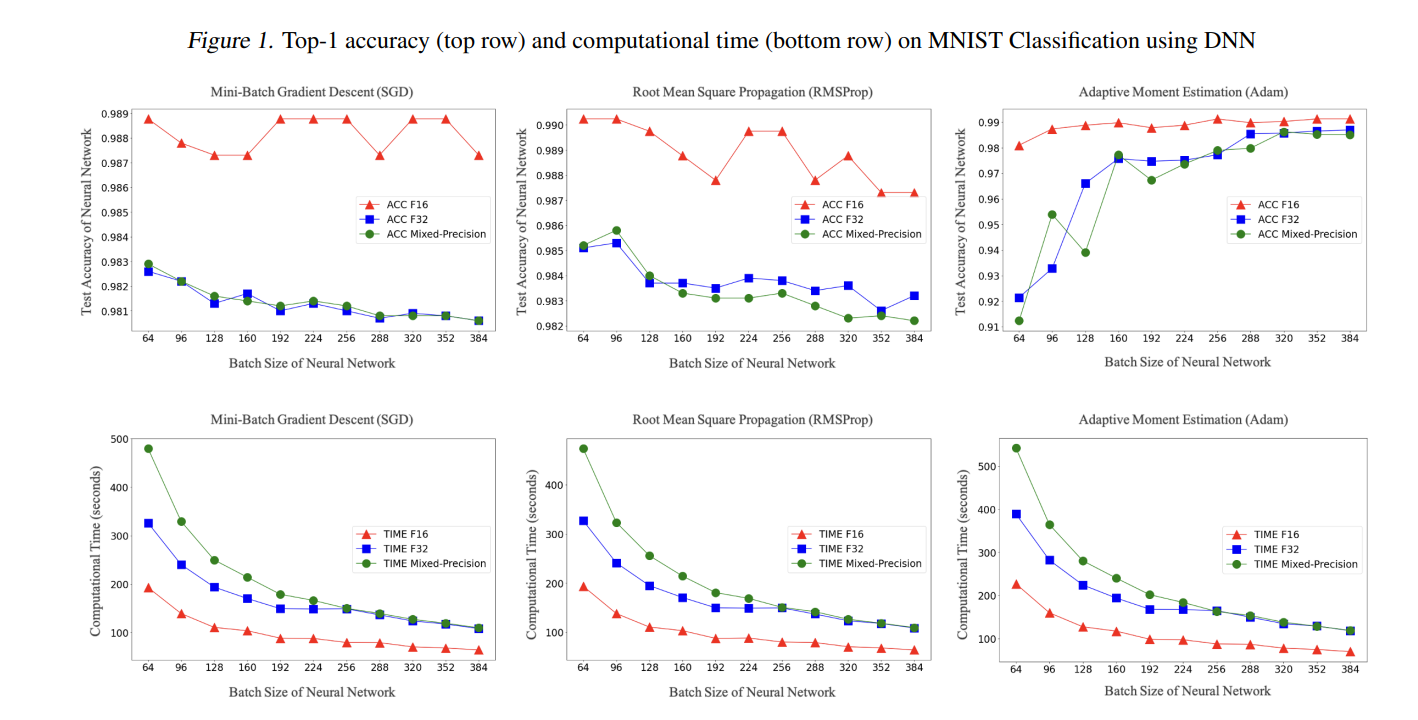
**实验设计：**

* 与pytorch原生算子的正确性对比



* 不同神经网络模型的推理准度：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 精度 \ 模型 | Resnet18 | Resnet32 | Resnet50 | Lenet | Transformer | …… |
| (8,23) |  |  |  |  |  |  |
| (5,10) |  |  |  |  |  |  |
| (5,2) |  |  |  |  |  |  |
| …… |  |  |  |  |  |  |



* 理论正确性的验证

训练/推理过程中统计各项数据(swamp/均值/方差/数据分布/vrr/v(n) )，代入公式中计算。得出结论