第三代英特尔® 至强® 可扩展处理器 (Ice Lake) 和英特尔® 深度学习加速助力阿里巴巴 Transformer 模型性能提升

Wanchen Sui
Minmin Sun
阿里巴巴集团

Feng Tian
Penghui Cheng
Changqing Li
Pujiang He
Haihao Shen
英特尔

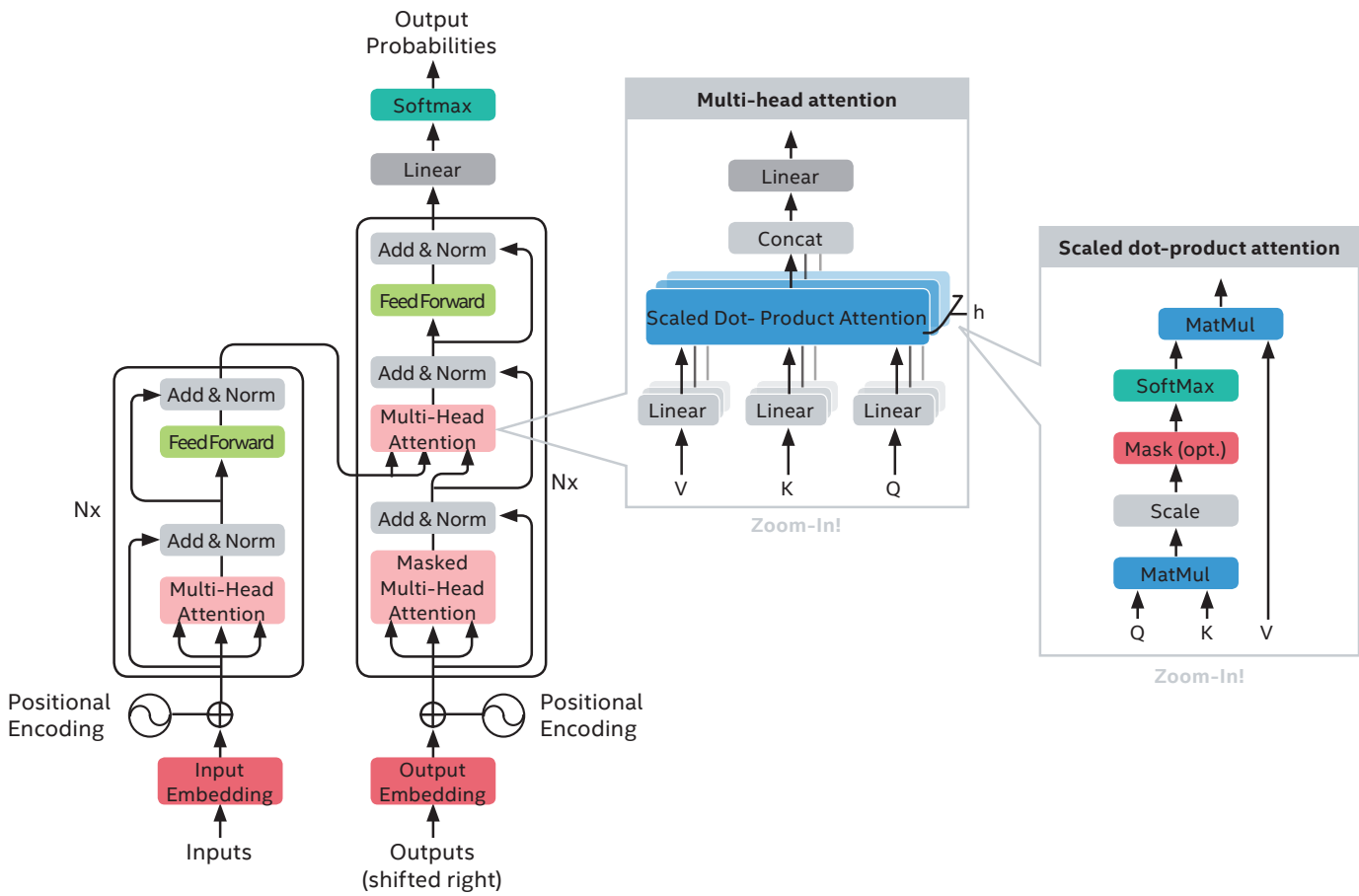
引言

第三代英特尔® 至强® 可扩展处理器采用了英特尔 10 纳米 + 制程技术。相比于第二代英特尔® 至强® 可扩展处理器，该系列处理器内核更多、内存容量和频率更高。阿里巴巴集团和英特尔的技术专家共同探索了这些能力对人工智能应用的意义，特别是在与英特尔® 深度学习加速 (英特尔® DL Boost) 结合使用时。我们还探索了英特尔® 低精度优化工具 (英特尔® LPOT)，助力客户在基于英特尔® 至强® 可扩展处理器的平台上快速开发和部署 AI INT8 模型。我们在第三代英特尔® 至强® 可扩展处理器上优化了阿里巴巴 Transformer 模型，并证明了 FP32 和 INT8 推理的性能相较于上一代处理器分别提升了 1.36 倍和 1.42 倍。

技术概览

Transformer 是阿里巴巴端到端 AI 机器学习平台 (PAI) 使用的关键模型，它广泛应用于自然语言处理 (NLP) 任务，并通过阿里巴巴线上服务供全球数百万用户使用。低时延、高吞吐量是 Transformer 成功的关键，而 8 位低精度操作正是有望满足这一要求的理想技术。

英特尔® DL Boost 为 8 位低精度推理的人工智能工作负载提供了强大的支持。依靠英特尔® LPOT 的支持，我们能在提升 8 位推理性能的同时显著降低精度损失。这些能力证明了英特尔在 AI 推理领域的领导地位，也展现了英特尔® DL Boost 和第三代英特尔® 至强® 可扩展处理器的强悍实力。



图一 Transformer 构造块的子图 (图片来源: Vaswani, et al., 2017)

模型分析

模型结构

图一展示了 Transformer 构造块的子图。

从此图可以看出，有些操作适用于 INT8 量化，以便更好地发挥采用英特尔® DL Boost 矢量神经网络指令的英特尔® AVX-512 (AVX512_VNNI) 的作用。我们利用英特尔® LPOT 自动生成一个符合预定义精度损失目标的 INT8 模型。目前，LPOT 支持在 PyTorch 原生 Imperative 路径上进行量化参数搜索调整。我们依靠它来探索所有可能的量化参数组合空间，例如每个可量化算子使用不同的每张量、每通道量化，以及非对称/对称设置，以便获得优化的量化模型。右图显示了使用英特尔® LPOT 生成 Transformer 低精度模型的代码片段。

```
import lpot
from lpot.metric import BaseMetric

class Dataset(object):
  def __init__(self):
    …
  def __getitem__(self, index):
    …
  def __len__(self):
    …

# Define a customized Metric function
class MyMetric(BaseMetric):
  def __init__(self, *args):
    …
  def update(self, predict, label):
    …
  def reset(self):
    …
  def result(self):
    …

# Quantize with customized dataloader and metric
quantizer = lpot.Quantization('./conf.yaml')
dataset = Dataset()
quantizer.metric = lpot.common.Metric(MyMetric)
quantizer.calib_dataloader = lpot.common.DataLoader(dataset, batch_size=1)
quantizer.eval_dataloader = lpot.common.DataLoader(dataset, batch_size=1)
quantizer.model = lpot.common.Model(model)
q_model = quantizer()
```

关于如何使用英特尔® LPOT 启用新的量化模型，更多详情参见 GitHub 的 LPOT 页面。

模型配置

阿里巴巴的 Transformer 模型是一个 PyTorch 模型。我们采用 profiling 的方法来分析模型性能。从下图中的 FP32 模型配置日志可以得知，它是一个计算密集型模型，在该模型中，总时间的 70% 均被计算密集型操作占用，如多项式乘 (conv) 和矩阵相乘 (matmul)。从中可知，AVX512_VNNI 指令能为 Transformer 模型带来显著的性能提升，而第三代英特尔® 至强® 可扩展处理器更高的内存带宽和频率也有利于内存密集型操作。

```
-----------------------  ------------  ------------  ...  ------------
                  Name    Self CPU %      Self CPU  ...   # of Calls
-----------------------  ------------  ------------  ...  ------------
               aten::mm        37.67%     76.644ms  ...          331
aten::mkldnn_convolution        29.32%     59.650ms  ...            2
            aten::copy_         3.84%      7.821ms  ...          664
             aten::bmm          2.84%      5.779ms  ...          144
            aten::add_          2.35%      4.791ms  ...          331
                forward         2.10%      4.277ms  ...            1
         aten::threshold        1.98%      4.038ms  ...           44
          aten:: softmax        1.93%      3.922ms  ...           72
```

以下是 INT8 模型配置日志，从中可以得知，如果对所有矩阵相乘 (matmul) 操作进行量化，计算性能可提升 76.644 / (20.296 + 6.632) = 2.84 倍。注：对多项式乘(conv)操作进行量化后，计算性能可提升 59.65 / 11.65 = 5.12 倍，超过了 4 倍理论峰值性能提升。这是因为 FP32 多项式乘 (conv) 操作在 oneDNN 路径上运行，与 INT8 多项式乘 (conv) FBGEMM 操作相比，该路径实际上包括实际多项式乘 (conv) 计算之前和之后的两个额外 reorder 算子。

```
----------------------  ------------  ------------  ...  ------------
                  Name    Self CPU %      Self CPU  ...   # of Calls
----------------------  ------------  ------------  ...  ------------
      quantized::linear        20.27%     20.296ms  ...          289
quantized::conv2d_relu        11.64%     11.653ms  ...            2
           aten::copy_         7.63%      7.636ms  ...          664
quantized::linear_relu         6.62%      6.632ms  ...           42
             aten::bmm         5.91%      5.913ms  ...          144
               forward         4.20%      4.208ms  ...            1
         aten:: softmax        3.92%      3.926ms  ...           72
```

性能与验证

我们分别在第二代和第三代英特尔® 至强® 可扩展处理器上测试了 Transformer 模型，均得到了显著的性能提升。FP32 和INT8 端到端性能提升分别如表一和表二所示。

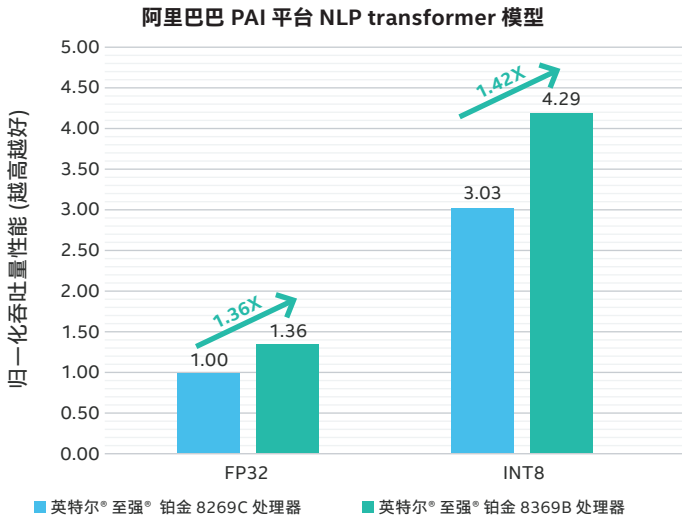
测试用例			第二代英特尔® 至强® 可扩展处理器 精度：FP32 框架：PyTorch 1.7.1 实例数：26	第三代英特尔® 至强® 可扩展处理器 精度：FP32 框架：PyTorch 1.7.1 实例数：32	性能增益
模型	数据集	批大小	吞吐量 (句/秒)	吞吐量 (句/秒)	百分比
Transformer	客户数据集	1	51.32	69.99	36%

表一 Transformer 模型 FP32 推理性能提升

测试用例			第二代英特尔® 至强® 可扩展处理器 精度：INT8 框架：PyTorch 1.7.1 实例数：26	第三代英特尔® 至强® 可扩展处理器 精度：INT8 框架：PyTorch 1.7.1 实例数：32	性能增益
模型	数据集	批大小	吞吐量 (句/秒)	吞吐量 (句/秒)	百分比
Transformer	客户数据集	1	155.39	219.96	42%

表二 Transformer 模型 INT8 推理性能提升

图二以图表形式显示了测试结果。



图二 基于 FP32 和 INT8 数据类型获得的代际性能提升

采用最新的英特尔® DL Boost (INT8) 技术后，性能得到了大幅提升，与 FP32 解决方案相比约提升 3.1 倍；在阿里巴巴定制的第三代英特尔® 至强® 可扩展处理器平台，总吞吐量与第二代英特尔® 至强® 可扩展处理器平台相比提高了约 42%。

精度方面，我们采用客户数据对 INT8 Transformer 模型进行验证，结果显示精度损失为 0.4%，能够满足客户需求。

测试用例			第二代英特尔® 至强® 可扩展处理器 精度：FP32	第三代英特尔® 至强® 可扩展处理器 精度：INT8	精度损失
模型	数据集	批大小	得分：(由客户评估项目报告)		
Transformer	客户数据集	1	91.45%	91.05%	0.4%

结论

与第二代英特尔® 至强® 可扩展处理器系列相比，第三代英特尔® 至强® 可扩展处理器提升了内核数量、频率和内存带宽，这令 PyTorch Transformer INT8 模型的性能提升了 1.42 倍，PyTorch Transformer FP32 模型的性能提升了 1.36 倍。阿里巴巴采用英特尔最新处理器和 INT8 量化工具后，可为阿里巴巴 PAI-Blade 推理工具集带来 3.1 倍性能提升。阿里云预计，这将有助于加快 Transformer 任务的运行，并向阿里巴巴数百万客户提供更高效的服务。

配置详情

基于 PyTorch 1.7.1 的阿里巴巴 PAI NLP Transformer 模型在第三代英特尔® 至强® 可扩展处理器上的吞吐量性能

基准配置： 英特尔截至 2021 年 3 月 19 日的测试。2 节点，2* 英特尔® 至强® 铂金 8269C 处理器，26 核，超线程开启，睿频开启，总内存 192GB (12 插槽 / 16 GB/ 2933 MHz)，BIOS: SE5C620.86B.02.01.0013.121520200651(0x4003003)，CentOS 8.3，4.18.0-240.1.1.el8_3.x86_64，编译器：gcc 8.3.1，Transformer 模型，深度学习框架：PyTorch 1.7.1，https://download.pytorch.org/whl/cpu/torch-1.7.1%2Bcpu-cp36-cp36m-linux_x86_64.whl，BS=1，客户数据，26 个实例 / 2 插槽，数据类型：FP32 / INT8

新配置： 英特尔截至2021年3月19日的测试。2 节点，2* 英特尔® 至强® 铂金 8369B 处理器，32 核，超线程开启，睿频开启，总内存 512 GB (16 插槽 / 32GB / 3200 MHz)，BIOS: WLYDCRB1.SYS.0020.P92.2103170501 (0xd000260)，CentOS 8.3，4.18.0-240.1.1.el8_3.x86_64，编译器：gcc 8.3.1，Transformer 模型，深度学习框架：PyTorch 1. 7. 1，https://download.pytorch.org/whl/cpu/torch-1.7.1%2Bcpu-cp36-cp36m-linux_x86_64.whl，BS=1，客户数据，32 个实例 / 2 插槽，数据类型：FP32 / INT8

所有性能数据均为实验室环境下测试所得。

了解更多

- 第三代英特尔® 至强® 可扩展处理器
- 阿里巴巴 AI 机器学习平台
- 英特尔® DL Boost
- 英特尔® LPOT