

OpenVINO推理性能优化

火狐狸

不时地挑战一下不可能

19 人赞同了该文章

OpenVINO最新版本是2021.3版本，自从2021.1版本以后，增加支持了C的API，这里还是以常用的C++的API来说明如何调优和加速。

这里简单以视觉模型来说明推理过程，和调优方法：

假设你已经完成模型转换，即将开源框架(如TF, Caffe)训练出来的模型转为IR格式。整个推理流程可以用下图来描述，下面根据这个流程来描述可以调优的方法和经验。

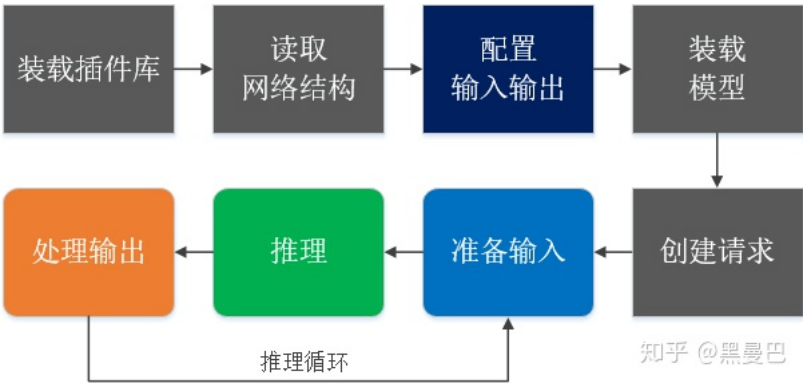


图1 推理流程

1. 装载插件库

```
InferenceEngine::Core core; // 管理处理器和扩展插件
```

2. 读取模型结构

```
auto network = core.ReadNetwork("model.xml");
```

3. 配置输入和输出

OpenVINO默认的通道顺序是BGR，在输入的时候，如果拿到的数据不是BGR格式，需要预处理通道顺序，这里通过setColorFormat接口进行调整。预处理不仅可以调整通道，还可以Resize算法类型，设置平均图(逐像素平均或逐通道平均)。

Enumerator	
RAW	Plain blob (default), no extra color processing required.
RGB	RGB color format.
BGR	BGR color format, default in DLDT.
RGBX	RGBX color format with X ignored during inference.
BGRX	BGRX color format with X ignored during inference.
NV12	NV12 color format represented as compound Y+UV blob.
I420	I420 color format represented as compound Y+U+V blob.

图2 图像通道类型

注意：如果是NV12或I420格式，不支持批量推理。

```
// 输入
InferenceEngine::InputsDataMap input_info(network.getInputsInfo());
/** Iterate over all input info**/
```




```
...
}
```

• 级联网络

从一个网络的输出获得Blob，输入下一个网络的输入Blob。

```
auto output = infer_request1->GetBlob(output_name);
infer_request2->SetBlob(input_name, output);
```

• 级联网络中处理ROI

当第一个网络的输出ROI是第二个网络的输入，无需重新为ROI结果分配内存，例如，当第一个网络检测视频帧上的对象时(存储为输入Blob)，第二个网络接收检测到的边界框(帧内的ROI)作为输入。在这种情况下，允许第二个网络重用预先分配的输入blob(由第一个网络使用)，并且只裁剪ROI，而不分配新的内存通过InferenceEngine::make_shared_blob()接口(参数为InferenceEngine::Blob::Ptr和InferenceEngine::ROI)。

```
/** inputBlob points to input of a previous network and
    cropROI contains coordinates of output bounding box */
InferenceEngine::Blob::Ptr inputBlob;
InferenceEngine::ROI cropRoi;
...
/** roiBlob uses shared memory of inputBlob and describes cropROI
    according to its coordinates */
auto roiBlob = InferenceEngine::make_shared_blob(inputBlob, cropRoi);
infer_request2->SetBlob(input_name, roiBlob);
```

分配适当类型和大小的Blob，然后将图像和数据输入到Blob中，通过InferenceEngine::InferRequest::SetBlob()接口设置到请求里。

```
/** Iterate over all input blobs */
for (auto & item : inputInfo) {
    auto input_data = item->second;
    /** Create input blob */
    InferenceEngine::TBlob<unsigned char>::Ptr input;
    // assuming input precision was asked to be U8 in prev step
    input = InferenceEngine::make_shared_blob<unsigned char>(InferenceEngine::SizeVect
    input->allocate());
    infer_request->SetBlob(item.first, input);
    /** Fill input tensor with planes. First b channel, then g and r channels */
    ...
}
```

7. 推理

• 异步模式

异步模式会立即返回结果，不会阻塞主线程，使用wait()等待推理结果。

```
infer_request->StartAsync();
infer_request.Wait(InferRequest::WaitMode::RESULT_READY);
```

Wait()有三种模式可用：

- 1) 指定阻塞最大时间，该方法会被阻塞，直到指定的超时过期或结果可用(以先出现的时间为准)。
- 2) InferenceEngine::InferRequest::WaitMode::RESULT_READY，一直等待，直到有推理结果出来。

3) InferenceEngine::InferRequest::WaitMode::STATUS_ONLY, 立即返回请求状态, 它不会阻塞或中断当前线程。



- 同步模式

```
infer_request->Infer();
```

8. 检查输出并处理结果

不推荐通过std::dynamic_pointer_cast将Blob到TBlob转换, 最好通过buffer()和as()来做。



```
for (auto &item : output_info) {
    auto output_name = item.first;
    auto output = infer_request.GetBlob(output_name);
    {
        auto const memLocker = output->cbuffer(); // use const memory Locker
        // output_buffer is valid as long as the lifetime of memLocker
        const float *output_buffer = memLocker.as<const float *>();
        /** output_buffer[] - accessing output blob data **/
    }
}
```

到此为止, 就完成了整个推理的API调用和相关配置。也期待你能以正确的姿势来使用OpenVINO, 性能得到一定得提升。

编辑于 2021-05-29 17:12

性能优化 硬件加速 加速

文章被以下专栏收录

- **AI架构与优化**
人工智能相关的软硬件知识和算法模型优化加速等技术
- **AI模型部署**
主要介绍AI模型的落地与部署, 关注AI芯片

推荐阅读



使用 c++ 来加速神经网络的推理(1)
zideajang



OpenVino 的安装及配置
塔叔

OpenVINO_Ubuntu安装



OpenVINO是Intel开源的一个工包, 主要用于计算机视觉方面的神经网络优化。它主要负责优化神经网络, 以便进行轻量级部署, 不含神经网络的训练。训练神经网络常用, MXNet, PyTorch, Tens

万岁爷

35 条评论

切换为时间排序

写下你的评论...



c++速度会更快一点，python上手容易，根据自己需求选择

👍 赞

 lili皮蛋 回复 火狐狸 (作者) 2020-09-23

你好，我是分类网络crnn，c++测infer_request->Infer()推理的时间比python版本的速度慢，请问您知道是什么原因吗，谢谢！

👍 赞

 火狐狸 (作者) 回复 lili皮蛋 2020-09-23

慢多少，指标贴一下

👍 赞

展开其他 1 条回复

 守一波宁静 2020-06-29

c++有点懵，我还是用python吧，mo.py对我来说已经够用了

👍 赞

 旗木卡卡西 2021-09-16


请您使用过级联网络那一栏吗？也就是SetBlob(input_name, output);使用其他网络的输出作为网络的输入，如果这里面网络的输出是256通道的，会不会报错啊？我这里是报错了，看起来好像是限制了网络的输入通道必须是1~4通道的

👍 赞

 一只小飞象 2021-07-19

您好，我有两个问题，对于计算棒重复插拔造成的错误，除了重启，您有其他方法解决吗？还有一个问题是，当网络采用异步处理时，当前帧返回的实际上是上一帧的处理结果，这对于实时视频流处理，会有一定的延时，请问异步操作适用于实时视频流处理吗？还是我操作有误？谢谢您

👍 赞

 君临天下 2021-07-14

请问被部署的设备上是不是也要安装openvino环境

👍 赞

 火狐狸 (作者) 回复 君临天下 2021-07-14

打包一下环境，就是几个so，和你的程序一起发布即可

👍 赞

 君临天下 回复 火狐狸 (作者) 2021-07-18

请问so是啥意思，小白刚入门😂

👍 赞

 总会学会 2021-05-27


新手学多久才能达到火老板的水平？

👍 赞

 火狐狸 (作者) 回复 总会学会 2021-05-29

加油学，很快就熟练了👍

👍 赞

 宝木三少DA先生 2021-04-23

您好，初始化模型时您选择的是U8数据格式，这个在转换模型时需要做一些特殊操作吗？

👍 赞

 火狐狸 (作者) 回复 宝木三少DA先生 2021-04-23

这个和模型相关，训练时用的U8，输入就是U8，训练时做了归一化，输入就是FP32，不知回答你的问题了没

👍 赞



您好,请问不推荐通过std::dynamic_pointer_cast将Blob到TBlob转换 这个说法有什么特别的来由么? 我看给的sample里使用的是dynamic_pointer_cast 这个方法....而且buffer 这个接口现在也不推荐了,推荐使用map

👍 赞



火狐狸 (作者) 回复 知乎用户

2021-03-26

估计是多个模型级联的时候, 内存上的效率考虑, 可以测一下

👍 赞



知乎用户 回复 火狐狸 (作者)

2021-03-26

还有请问下 input 层Layout 顺序设置成NCHW,是不是输入的inputBlob 也一定要是NCHW?设定成NHWC opencv会自动转换么?

👍 赞

展开其他 1 条回复



汪汪

2021-02-01

您好, 请教一个问题, 我用原生tf环境做训练, 但用avx2编译的加速tf环境做部署, 发现openvino并没有比avx2的tf快, 想问一下这样情况该怎么解决? 是不是infer的配置不对, 导致没有发挥openvino最佳的性能? 期待回答

👍 赞



火狐狸 (作者) 回复 汪汪

2021-02-02

什么模型? 什么cpu? 用了多少core?

👍 赞



汪汪 回复 火狐狸 (作者)

2021-02-04

一个推荐模型, 类似于wide&deep, cpu是服务器的intel gold 5128 @2.3ghz, 16核, 又试了几天, 还是无法更快。openvino我编译的时候关了cldnn, 看编译配置默认开启了avx2

👍 赞

展开其他 1 条回复



逐云者

2020-11-06

你好, 之前用tf+mkl, 多线程的情况下, 单次推理确实会很快, 但是并发时表现很差, 甚至不如单线程的情况, 请问openvino如何调优来避免类似问题呢?

👍 赞



火狐狸 (作者) 回复 逐云者

2020-11-07

这个问题主要看模型, 是CNN的, 还是RNN, 前者应该会快, 后者单线更快, openvino只不过集成了生产-消费者模式, 维护一个推理队列, 异步并发推理

👍 赞



清风生

2020-09-27

你好, 我看openvino兼容tensorflow serving, 所以就用了java的grpc。但是java的grpc会有很多时间耗费在输入数据反序列化 (input deserialization), 请问您这边有什么好的优化思路吗。比如shape(1,3,224,224)的图像, python需要1ms, java需要37ms。

👍 赞



火狐狸 (作者) 回复 清风生

2020-09-27

你的java客户端连接的服务端慢, 是tf-serving慢, 还是OpenVINO Model Server慢?

👍 赞



清风生 回复 火狐狸 (作者)

2020-09-28

OpenVINO Model Server慢, 看日志会执行一个input deserialization过程, 耗时比较长。在经过openvino的IR格式的转换之后, 用java客户端调用openvino model sever反而比tf-serving慢。python则没有这个问题。

👍 赞



南城北巷

2020-09-14

请问如果需要应用多个模型文件具体应该这么处理呀？

👍 赞



Hansansui

2021-12-07

大哥，模型加载那个线程数怎么设置啊，能给个config的实例吗？

👍 赞



火狐狸 (作者) 回复 Hansansui

2021-12-08

设置一下KEY_CPU_THREADS_NUM即可

👍 赞



Hansansui 回复 火狐狸 (作者)

2021-12-08

```
std::map<std::string, int> config ={
{InferenceEngine::PluginConfigParams::KEY_CPU_THREADS_NUM, 4} };
这样写有什么问题吗？编译报错。
```

👍 赞

展开其他 3 条回复