实践教程｜pytorch怎么使用c++调用部署模型?

CV开发者都爱看的 极市平台 2023-05-17 20:01:25 发表于广东 手机阅读

作者丨Civ@知乎（已授权）

来源丨https://www.zhihu.com/question/66532235/answer/2782357337

编辑丨极市平台

极市导读

本文以C++推理框架ncnn为例，介绍一下部署的大致流程。其它C++推理框架的思路类似，唯一的学习成本是推理框架本身的API >>加入极市CV技术交流群，走在计算机视觉的最前沿

方法有很多种，比较简单的路径是：

PyTorch模型 --> ONNX格式 --> C++推理框架

本文以C++推理框架ncnn为例，介绍一下大致流程。其它C++推理框架的思路类似，唯一的学习成本是推理框架本身的API。

一、PyTorch模型转ONNX

> ONNX is an open format built to represent machine learning models. ONNX defines a common set of operators - the building blocks of machine learning and deep learning models - and a common file format to enable AI developers to use models with a variety of frameworks, tools, runtimes, and compilers.

简单来说，可以把ONNX当做一个中间格式。绝大多数的机器学习/深度学习框架都可以将自身的模型转换成ONNX，同样也能把ONNX转换成自身框架的格式，如下图所示。

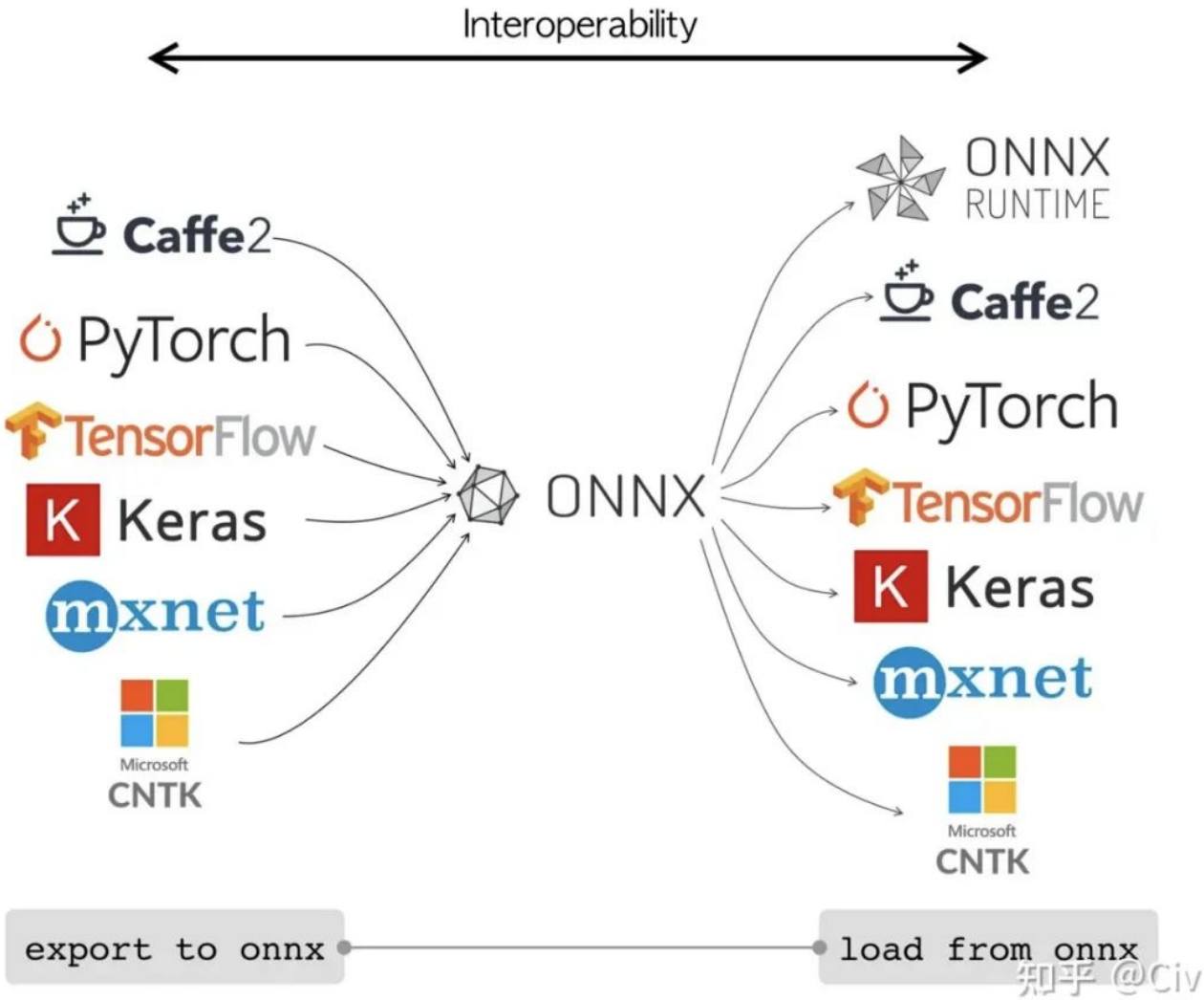


图1 不同框架的模型利用ONNX进行相互转换

ONNX官网地址：https://onnx.ai/

在PyTorch中，可以用如下方法非常方便地将一个PyTorch模型存储为ONNX格式：

```
import torch

# 指定输入尺寸，ONNX需要这个信息来确定输入大小
# 参数对应于 (batch_size, channels, H, W)
dummy_input = torch.randn(1, 3, 224, 224, device="cuda")

# model为模型自身
# dummy_input根据自己的需求更改其尺寸
# "model.onnx"为输出文件，更改为自己的路径即可
torch.onnx.export(model, dummy_input, "model.onnx")
```

torch.onnx.export还有一些额外的参数可以实现更灵活的使用方法，详见https://pytorch.org/docs/stable/onnx.html。本文的示例足以让您能够成功部署自己的模型。

需要注意的是，ONNX的目的是“通用”，所以难免会在一些情况出现算子不兼容的情况。具体的表现是，当你把某个框架（例如PyTorch）的模型转成ONNX后，再将ONNX转成另一框架模型（例如ncnn）时，可能会报错（xxx算子不支持）。不兼容的情况多种多样，这里不举例说明了，需要具体情况具体分析。

一些有效的解决方法：

1. 使用ONNXSIM对ONNX模型进行精简。非常有效。个人建议：只要使用了ONNX，都用ONNXSIM对ONNX模型进行处理一次。Github地址：https://github.com/daquexian/onnx-simplifier。使用非常方便，使用“pip install onnxsim”安装，然后使用命令“onnxsim input_onnx_model_path output_onnx_model_path”即可。代码中调用也很简单，参考Git地址里的示例。

2. 避免依赖于中间变量的尺寸来进行运算。比如，在一些Image to Image的任务中，可能会根据中间tensor的尺寸来对另一些tensor进行resize。这时我们的做法是先去获取中间tensor的尺寸H、W，然后将它们作为参数送给其它方法。当遇到这种运算时，ONNX似乎会创建两个与H、W相关的变量，但它们的值会绑定为用dummy_input去forward一次时得到的H、W。这个值一旦绑定就不会改变。所以后续当使用不同尺寸输入时极大概率会报错（这点没有仔细验证过，但看中间结果很像是这种情况）。

另外强烈建议使用一些网络可视化工具。当遇到模型转换报错时可以用来方便定位出错的位置。个人比较喜欢的是netron，地址：https://github.com/lutzroeder/netron

放一张仓库中的图，效果如下：

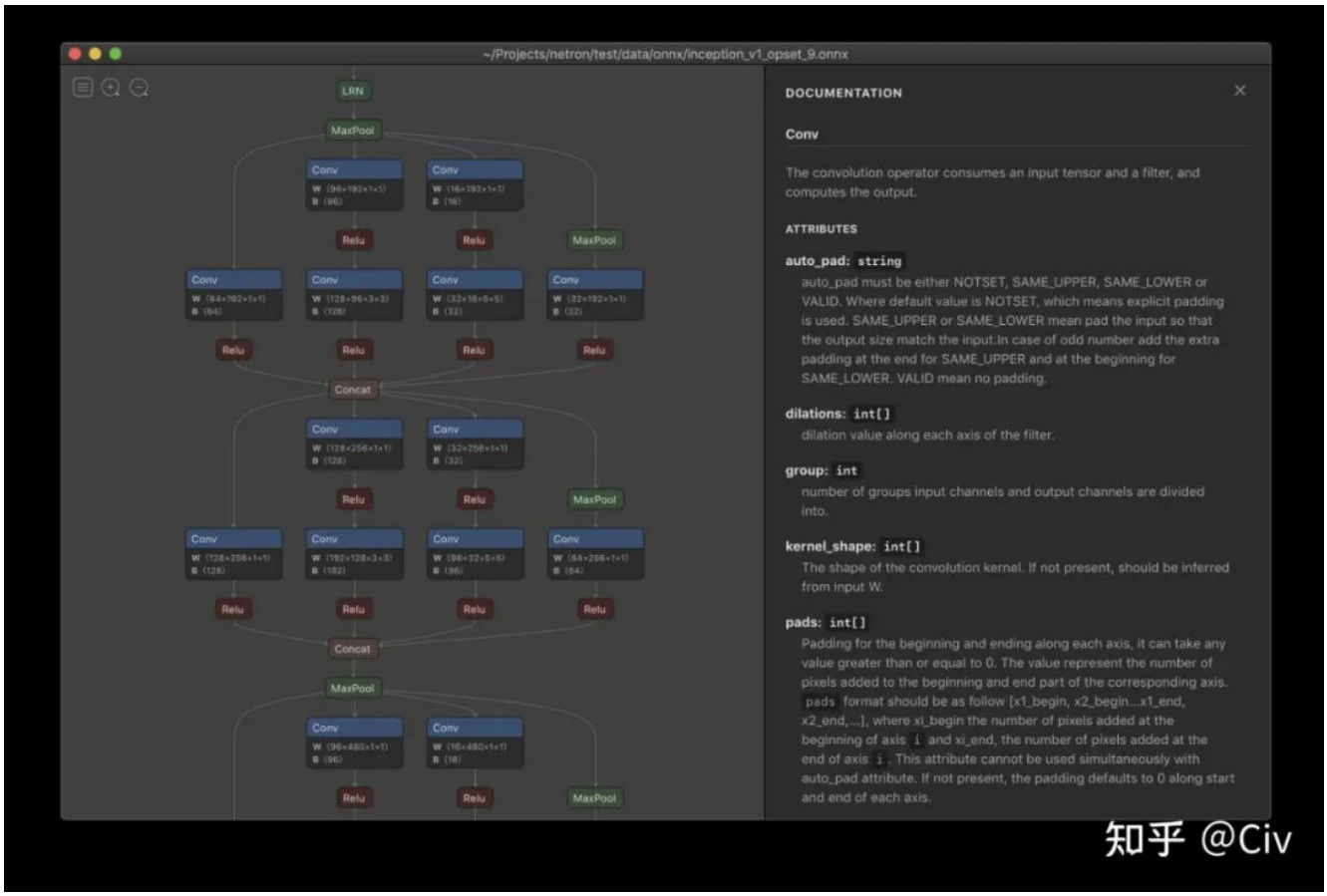


图2 netron效果图

二、ONNX转ncnn

ncnn是腾讯开源的轻量级推理框架。简单易用是它最大的特点。但当功耗、时耗是主要考虑点的时候，需要多尝试其它框架，如TensorFlow Lite。

ncnn地址：https://github.com/Tencent/ncnn

ncnn提供了将onnx转换为ncnn格式的工具。可以在此处找到：https://github.com/Tencent/ncnn/releases。例如，在Windows下，可以下载https://github.com/Tencent/ncnn/releases/download/20221128/ncnn-20221128-windows-vs2017.zip。解压后在x64或x86的bin文件夹中能够找到onnx2ncnn.exe。在命令行中使用如下命令即可将onnx转换为ncnn格式：

```
onnx2ncnn.exe onnx_model_path [ncnn.param] [ncnn.bin]
```

onnx_model_path 替换为自己的onnx模型地址。后两个参数可选。如果不写，那么会在onnx2ncnn.exe同目录下产生转换后的ncnn模型文件：一个.param文件和一个.bin文件。也可以自己填后两个参数来自己指定文件输出路径。

三、在ncnn下进行模型推理

在任何框架下推理都只需要两步：加载模型和将数据转化为框架格式。

ncnn下加载模型的方法为（还有其它方法）：

```
ncnn::Net model;  // 定义一个模型
model.load_param("model.param");    // 加载模型的param文件
model.load_model("model.bin");         // 加载模型的bin文件
```

加载模型后，只需要将数据转化为ncnn的格式即可。ncnn模型输入的格式是ncnn::Mat。

OpenCV的Mat转ncnn::Mat的方法全列于此处：

https://github.com/Tencent/ncnn/wiki/use-ncnn-with-opencv

如：

```
// cv::Mat a(h, w, CV_8UC3);
ncnn::Mat in = ncnn::Mat::from_pixels(a.data, ncnn::Mat::PIXEL_BGR2RGB, a.cols, a.rows);
```

在JNI中要将一个android bitmap转换为ncnn::Mat可参考官方示例：https://github.com/nihui/ncnn-android-squeezenet/blob/master/app/src/main/jni/squeezencnn_jni.cpp

代码如：

```
// ncnn from bitmap
ncnn::Mat in = ncnn::Mat::from_android_bitmap(env, bitmap, ncnn::Mat::PIXEL_BGR);
```

有了模型和输入，最后forward一次，再取结果即可：

```
ncnn::Extractor ex = model.create_extractor();

// input_name 可以通过netron对.param或.bin文件进行查看
// 将input_name替换为模型的第一个输入位置的名字即可
 ex.input(input_name, in);

ncnn::Mat out;  // 用来存放输出结果

// output_name可以通过netron对.param或.bin文件进行查看
// 将output_name替换为模型的输出位置的名字即可
ex.extract(output_name, out);
```

写在最后

只要是转换模型，大多数路径都是如此，学习成本并不高。主要是学习推理框架的成本。芯片厂商自身的推理框架相对复杂点，各种奇奇怪怪的条条框框。

极市干货

极视角动态：推进智能矿山建设，极视角「皮带传输系列算法」保障皮带安全稳定运行！

CVPR2023：CVPR 2023｜21 篇数据集工作汇总（附打包下载链接）

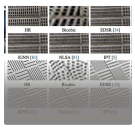
数据集：垃圾分类、水下垃圾/口罩垃圾/烟头垃圾检测等相关开源数据集汇总｜异常检测开源数据集汇总｜语义分割方向开源数据集资源汇总

阅读原文

喜欢此内容的人还喜欢

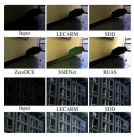
ICCV 2023｜南开程明明团队提出适用于SR任务的新颖注意力机制（已开源）

极市平台



ICCV23｜将隐式神经表征用于低光增强，北大张健团队提出NeRCo

极市平台



ICCV2023｜AlignDet：在各种检测器的所有模块实现无监督预训练

极市平台