**TVM总结**

1. **编译安装**

工具：gcc cmake llvm

1. 克隆源码

git clone --recursive https://github.com/apache/incubator-tvm tvm

1. 拷贝config.cmake

mkdir build

cp cmake/config.cmake build

1. 修改编译选项

针对不同硬件，加速库，编译器对config.cmake进行选择性修改，例如使用cuda，cuda选项设置‘ON’；注意LLVM选项设置ON后如果找不到LLVM，需要手动设置LLVM，将LLVM路径添加进去set(USE\_LLVM /path/to/llvm/bin/llvm-config)

4.编译

cd build

cmake ..

make -j4

5.安装

方法一：将TVM路径添加环境变量

修改~/.bashrc 添加如下内容

export TVM\_HOME=/path/to/tvmexport PYTHONPATH=$TVM\_HOME/python:$TVM\_HOME/topi/python:${PYTHONPATH}

方法二：利用setup.py安装

cd python; python setup.py install --user; cd ..

cd topi/python; python setup.py install --user; cd ../..

7.安装依赖库

pip3 install --user numpy decorator attrs

pip3 install --user tornado psutil xgboost

pip3 install --user mypy orderedset antlr4-python3-runtime

1. 测试

完成步骤1-6后进入python，如果可以导入tvm（import tvm），则编译安装成功。

**二．模型转换TVM**

目前tvm支持mxnet,keras,onnx,tensorflow,caffe2,darknet,TFlite框架模型的直接转换，pytorch需要先转onnx才可以转换tvm模型。

步骤一:模型提取（ONNX为例，其它框架类似）

import tvm

import onnx

from tvm import relay

onnx\_model = onnx.load('./resnet/model.onnx')

input\_name = 'input'

shape\_dict = {input\_name: (1, 3, 224, 224)}

mod, params = relay.frontend.from\_onnx(onnx\_model, shape\_dict)

步骤二:模型保存

with relay.build\_config(opt\_level=3):

graph, lib, params = relay.build\_module.build(

mod, target=target, params=params)

graph\_json\_path = "./onnx\_untuned.json"

with open(graph\_json\_path, 'w') as fo:

fo.write(graph)

param\_path = "./onnx\_untuned.params"

with open(param\_path, 'wb') as fo:

fo.write(relay.save\_param\_dict(params))

lib.export\_library("onnx\_untuned.so")

**三．模型自动调优**

目前tvm针对硬件自动调优支持x86 CPU,ARM CPU,NVIDIA GPU,MOBILE GPU

以NVIDIA GPU为例，注意事项如下：

1. 根据显卡类型设置target，如1080Ti显卡

target = tvm.target.cuda(model='1080ti')

如果使用cudnn

target = 'cuda -model=1080ti, -libs=cudnn, cublas'

1. 调优方式可以在本地机器调优，也可以局域网内主从机集群方式，后者可以多台机器同时调优加速调优速度。

主从机调优，需要执行如下操作：

主机：python -m tvm.exec.rpc\_tracker --host=0.0.0.0 --port=9190

从机：python -m tvm.exec.rpc\_server --tracker=主机IP地址:9190 --key=1080ti

主机： python -m tvm.exec.query\_rpc\_tracker --host=0.0.0.0 --port=9190

如果有如下类似输出，则联机成功

# .. code-block:: bash

# Queue Status

# ----------------------------------

# key total free pending

# ----------------------------------

# 1080ti 1 1 0

# ----------------------------------

调优选项配置如下

tuning\_option = {

'log\_filename': log\_file,

'tuner': 'xgb',

'n\_trial': 2000,

'early\_stopping': 600,

'measure\_option': autotvm.measure\_option(

builder=autotvm.LocalBuilder(timeout=10),

#runner=autotvm.LocalRunner(number=20,repeat=3,timeout=4,min\_repeat\_ms=150),#本地调优方式

runner=autotvm.RPCRunner(

'1080ti','0.0.0.0',9190,number=20,repeat=3,timeout=4,min\_repeat\_ms=150)

),#主从机调优方式

}

1. **Pytorch转TVM流程及注意事项**

**流程：**Pytorch--->onnx--->tvm

1. 安装最新版pytorch（新版pytorch对onnx功能支持更友好）
2. Pytorch模型转onnx

torch\_model = torch.load("save.pth") # pytorch模型加载

batch\_size = 1 #批处理大小

input\_shape = (3,244,244) #输入数据

x = torch.randn(batch\_size,\*input\_shape) # 生成张量

export\_onnx\_file=’test.onnx’

torch.onnx.export(torch\_model,x,export\_onnx\_file)

input\_names=["input"]

output\_names=["output"])

1. onnx转TVM

具体流程及代码示例参照**二.模型转换 TVM**

最终onnx模型转换tvm，文件有三个，分别为onnx\_untuned.json，onnx\_untuned.json，onnx\_untuned.so。

注意事项：

1.直接通过pytorch输出的onnx模型转tvm存在数据类型错误

例如会出现如下类似错误

concatenate(%451)an internal invariant was violated while typechecking your program [15:21:56] /home/zgy/tvm/src/relay/op/tensor/transform.cc:204: Check failed: e\_dtype == dtype (int64 vs. int32) : relay.concatenate requires all tensors have the same dtype

TVM默认的整形类型与ONNX中不一致导致，可以通过更改模型数据类型解决，或者采用下面方法解决。

利用该工具调整模型

pip install onnx-simplifier

python -m onnxsim input\_onnx\_model output\_onnx\_model

1. 输入名称不一致

转换onnx及tvm时如果‘input’不一致会报错，根据报错内容修改输入名称

1. 目前pytorch模型中有些op onnx不支持

RuntimeError: ONNX export failed: Couldn't export operator aten::adaptive\_avg\_pool2d

例如上面报错是因为pytoch 中adaptive\_avg\_pool2d自适应平均池化不适用于onnx，解决方式修改adaptive\_avg\_pool2d为通用avg\_pool2d。

当然还存在其它类型op onnx尚未支持