ONNX Development Environment

Chin Huang, Cognitive OpenTech

Data and AI
Open Source Dojo

Outline

- Overview
- System setup
- ONNX build from source
- Tensorflow install
- ONNX Tensorflow converter
  - Onnx-tensorflow build
  - Quick verification

Overview

- The instructions in the following slides are to set up the development environment for the ONNX Tensorflow converter. Additional information can be found at https://github.com/onnx/onnx-tensorflow
- We will go over the build process for the key converter dependencies ONNX and Tensorflow, but will not go into the development details for them, as additional details can be found, https://github.com/onnx/onnx and https://github.com/tensorflow/tensorflow, respectively.

Development dependencies

- System setup and packages
- ONNX master
- Tensorflow 2.1
- ONNX-Tensorflow master

System setup and packages

- Python3: The following instructions assume *python –V* returns python 3.6.x. The recommendation is to use virtualenv as the system build-in python3 is somewhat broken and needs additional patch work.

  sudo pip install virtualenv (or sudo pip3 install virtualenv)
  virtualenv venv_py3
  virtualenv -p /usr/bin/python3 venv_py3
  source venv_py3/bin/activate

- Git (should already installed during the git hands-on session)
- cmake (*sudo apt install cmake*)
- protobuf-compiler libprotoc-dev (*sudo apt install protobuf-compiler libprotoc-dev -y*)
- Verify: *python -V* should returns 3.x.x
- Verify: *dpkg -l* should show others are installed

ONNX

Build from source

- *git clone https://github.com/onnx/onnx.git*
- *cd onnx*
- *git submodule update --init --recursive*
- *pip install -e .* (instead of *python setup.py install* which is documented in ONNX readme because we need to know the directory in providing ONNX-Tensorflow development support status)

Verification and test

- change directory out of onnx
- *python -c "import onnx“*
- *pip install pytest nbval*
- change directory to onnx
- run *pytest*

Tensorflow

Use the stable 2.x release

- The Tensorflow master can be built manually but we use the latest release for stability
- *pip install -U tensorflow*
- *pip install -U tensorflow-addons*
- Now Tensorflow 2.x stable release is ready

Verification and test

- *python*
- >>> *import tensorflow as tf* (If you see ModuleNotFoundError: No module named 'google.protobuf', exit python with exit(), then uninstall and reinstall protobuf using pip)
- >>> *tf.__version__* returns '2.1.0'
- >>> *tf.add(1, 2).numpy()* returns 3, ignore system warnings if any

ONNX-Tensorflow

ONNX-Tensorflow dependencies

- Python3 (slide 5)
- ONNX (source build from master, slide 6)
- Tensorflow (latest stable 2.x release, slide 7)

Build from source

- *git clone https://github.com/onnx/onnx-tensorflow.git*
- *cd onnx-tensorflow*
- *pip install -e .*

Additional information can be found at *https://github.com/onnx/onnx-tensorflow.git*

ONNX-Tensorflow

Verification and test

- *python -c "import onnx_tf"* should not return errors other than warnings
- *python test/backend/test_model.py* (quickly run the model test)
- *python util/get_version.py* (should see something below)

```
Python version:
3.6.9 (default, Nov  7 2019, 10:44:02)
[GCC 8.3.0]
ONNX version:
1.7.0
ONNX-TF version:
1.5.0
Tensorflow version:
2.1.0
```

ONNX-Tensorflow

Additional setup for code format and analysis (as a reference, not used in the labs)

- Format code with yapf
  - *pip install yapf*
  - *yapf -rip --style="{based_on_style: google, indent_width: 2}" $FilePath$*
- Use pylint to check and analyze python code
  - *pip install pylint*
  - *wget -O /tmp/pylintrc https://raw.githubusercontent.com/tensorflow/tensorflow/master/tensorflow/tools/ci_build/pylintrc*
  - *pylint --rcfile=/tmp/pylintrc myfile.py $FilePath$*