How to run FakeLowP vs Glow tests

This was tested on Ubuntu 16.04 LTS but should work in general Linux system. The tested compiler is Clang-8.

Build Glow Onnxifi Library

Follow https://github.com/pytorch/glow/blob/master/README.md to install the dependency of Glow. Then at glow root run

```
mkdir build && cd build
cmake -G Ninja -DGLOW_BUILD_ONNXIFI_DYNLIB=ON ..
ninja all
```

Note that here you probably want to add other flags like `-DGLOW_WITH_NNPI=1` to enable specific backend if you have the flow set up. Also, make sure you have the LD_LIBRARY_PATH set correctly pointing to libomp.so path when compiling with -DGLOW_WITH_NNPI=1.

```
export LD_LIBRARY_PATH=/usr/lib/llvm-8/lib
```

Once built successfully, you will get an dynamic library at `build/lib/Onnxifi/libonnxifi.so`. We will use it later.

Build and Install PyTorch

Follow https://github.com/pytorch/pytorch/blob/master/README.md to install the dependency of PyTorch. It might be easy to setup a python virtualenv or conda. And please use Python > 3.5.2 because hypothesis library will expose a bug in Python which is fixed after 3.5.2. Something like 3.7 might be good enough. You can install python3.7 with

```
sudo apt-get install -y build-essential checkinstall libreadline-gplv2-dev libncursesw5-dev
wget https://www.python.org/ftp/python/3.7.4/Python-3.7.4.tgz && tar -xf Python-3.7.4.tgz
cd Python-3.7.4
./configure && make -j 8 && sudo make altinstall
```

Once you installed Python 3.7, here I give a virtualenv flow:

```
sudo pip3.7 install virtualenv
python3.7 -m venv venv3
source venv3/bin/activate
cd pytorch
pip install -r requirements.txt
pip install pytest hypothesis protobuf
```

You probably need to install gflags-dev too with

```
sudo apt-get install libgflags-dev
```

Once you have all the dependency libs installed, build PyTorch with FakeLowP op support

```
USE_CUDA=0 USE_ROCM=0 USE_FAKELOWP=ON DEBUG=1 CMAKE_BUILD_TYPE=Debug USE_GFLAGS=1 USE_GLOG=1
```

The key options here are `USE_FAKELOWP=ON` which enables building of FakeLowP operators and `USE_GFLAGS=1` which enables gflags as we use gflags in Glow to pass options. Other flags are mostl for fast build time and debug purpose.

Run the test

You can now run the tests with command like the following when you are inside the virtual python env:

```
OSS_ONNXIFI_LIB=${PATH_TO_GLOW}/build/lib/Onnxifi/libonnxifi.so pytest pytorch/caffe2/contri
```