Snapdragon NPE Support

Build

Use the typical build_android script, but include a couple Cmake options to enable Snapdragon NPE:

```
NPE_HEADERS=/path/to/snpe-1.2.2/include/
NPE_LOCATION=/path/to/snpe-1.2.2/lib/arm-android-gcc4.9/libSNPE.so
./scripts/build_android.sh -DUSE_SNPE=ON -DSNPE_LOCATION=$NPE_LOCATION -
DSNPE_HEADERS=$NPE_HEADERS
```

this will enable the Snapdragon NPE Operator, which is a Caffe2 operator that can execute NPE `dlc` files.

Usage

Follow Qualcomm's instructions to convert a model into `dlc` format. You can then use the `dlc` as a Caffe2 operator in Python:

```
with open('submodel.dlc', 'rb') as f:
    dlc = f.read()

op = core.CreateOperator('SNPE', ['data_in'], ['data_out'],
         arg=[
             utils.MakeArgument("model_buffer", dlc),
             utils.MakeArgument("input_name", "data") # Assuming DLC's first layer
takes in 'data'
         ]
     )
```

and adding the operator to your model as you would normally.

Debug

`libSNPE.so` is a shared library that is loaded at runtime. You may need to specify the location of the library on your Android device when running standalone binaries. The runtime assumes it will be able to `dlopen()` a file named `libSNPE.so` at the location specified by `gSNPELocation()`. Either change this value at runtime or use an environment variable such as `LD_LIBRARY_PATH`.

You also need `libgnustl_shared.so` from Android NDK to be loaded in order to run standalone binaries.