Using the Tensorflow C library on Blue Waters with the bwpy environment

When logging onto Blue Waters:

- Use the **module load bwpy** and **bwpy-environ** commands to set up the **bwpy environment**

The Tensorflow C Library is located in: /mnt/bwpy/single/usr/include/tensorflow/c:

- In the C folder, one will find the Tensorflow **c_api.h**
  - Through what I could find, the Tensorflow C library on bwpy does not provide machine learning functions/ops that Tensorflow Python and Tensorflow C++ have. For example:
    - tf.matmul (Tensorflow Python)
    - tensorflow::ops::SparseReshape (Tensorflow Core)
  - This is why the Tensorflow **Core** Library is useful for pure c/c++ Tensorflow programs.
    - Tensorflow **Core** (C++)provides machine learning functions through a library of ops that the c_api.h can use in graph/model creation.

With only the **c_api.h** provided by the **bwpy environment**:

- Given only the **c_api.h** we can still train an exported model created through a higher level Tensorflow language such as Tensorflow Python.
  - For this project, a basic MNIST Tensorflow Python script is used to export a .pb model. From there, the c_api.h is used to load the .pb model and then to train it with the downloaded MNIST data from http://yann.lecun.com/exdb/mnist/

To compile/run the Tensorflow C program:

- run the **genM.py** with the default **python2** on bwpy to create the **graph.pb**
- Use the **module swap PrgEnv-cray PrgEnv-gnu** commands to get an up to date gcc compiler that will be able to compile Tensorflow C.
- Run the **Makefile**
- Use **./train.exe**

Readme, Files, and Resources for this project are in **gitlab**:

- https://git.ncsa.illinois.edu/garywang/tensorflow-c-on-bwpy

Takeaways:

- Ultimately, the Tensorflow **c_api.h** is not enough on its own to create and train a model without the help of other Tensorflow libraries. The **c_api.h** is used as a "pipe lining" api to provide Tensorflow libraries like Tensorflow **Core** with a graph/model framework where Users can:
  - Input their custom/provided ops into.
  - Input prebuilt models into for training.
- Combined with a higher level Tensorflow language to pre-build a model, a User can still be able to avoid bulky compilation/execution (Python) by using a lightweight executable to train the model as produced by a Tensorflow C/C++ program.
- If Users on Blue Waters want to do pure Tensorflow C++ programming, Tensorflow **Core** should be included in the next Tensorflow build. (If already included, make it easier to access/find)

2:35 PM