STATUS UPDATE AND FURTHER STEPS - Feb-15 to March-4

1) C++ CUDA extension code for convolution to be integrated with the forward() in python:

    a) Draft version of custom_conv operation written in c++ making use of the ATen, THC libraries to bring tensor functionality to c++
    b) Further implementations: Replace the *torch::mm* function in the code to the custom MBM implementation.
    c) *Code attached along with the mail*.

2) Work to set up the environment for CUDA development:
    a) Tried several approaches to set up the environment to make the nvcc compiler (nvidia cuda compiler) visible to the working environment. This took up most of the time as the documentation is not very accessible. My approaches have been given in detail below:
        i) Conda Environment Setup

        Creating a virtual environment using the conda - a virtual env/ package manager.
        The following were the packages and dependencies downloaded:
        -> python - 3.7.10
        -> pytorch - 1.7
        -> torchvision - 0.8
        -> cudatoolkit - 10.2.89 (CUDA - 10.2) via (*I) - to install on conda env
        -> cudnn - 7.6 9 (CUDA - 10.2)
        -> scikitlearn
        -> matplotlib
        -> NVidia Driver Version - 440 (CUDA - 10.2)
        -> libtorch - pre-cxx11 ABI build for CUDA - 10.2

        The conda environment doesn't install the nvcc when *cudatoolkit* package is installed. This is an issue yet to be fixed. The alternative approach is to to install the *nvcc_linux-64* package from *conda-forge* channel to make the conda env aware of the cuda-nvcc available outside the conda environment:
        -> nvcc_linux-64=10.1
        -> cudatoolkit - 10.2.89 via (*II) - to install on local machine

        *(*I) conda install cudatoolkit=10.2*
        *(*II) wget https://developer.download.nvidia.com/compute/cuda/10.2/Prod/local_installers/cuda_10.2.89_440.33.01_linux.run*
        sudo sh cuda_10.2.89_440.33.01_linux.run

It is important to be able to detect the nvcc to be able to compile the Pytorch C++ API.

Among various other combinations of specifications, this environment build was able to detect nvcc,compile and run basic cuda programs. But when trying to bind it to Pytorch libraries, there were library errors which have not been referenced in any of the forums. So I did not get a clear solution to the problem.

ii) **Pip Environment Setup**

The idea was to make a virtual environment using the pip virtualenv package that manages packages/ env to work on.

The setup was similar to the one done for the conda environment. But I was getting a linker error, when trying to compile codes using Pytorch C++ APIs. Main error that occurred was: **Ld cannot find entry symbol _start; not setting start address**. There were no fixes that could correct this error, so I had to look for other options.

iii) **Docker Image - Nvidia NGC Containers**

The Nvidia NGC containers contain all the essential libraries and packages required as a single docker image which can be pulled from their repositories and used.

I tried pulling the Pytorch+CUDA docker image, but the file size is just too big for the root directory to hold.

3) **Next Steps:**

a) I will be expanding the root dir and try to implement it once using the Pytorch+CUDA framework NGC container to make it work. Any suggestions and directions on this would be really helpful.

b) I will also parallely try a final build using conda env which I haven't tried out, using the latest Nvidia Driver - version 11.0, Libtorch(for cuda 11.0) and test it out. If there is a build in conda env or any other env that you can suggest, would be really helpful.

c) Implementing MBM algo in C++ and integrating with the draft custom_conv operation written using Pytorch C++ API and binding it to the python wrapper.

4) **REFERENCES:**

a) Useful links:

i) https://chrischoy.github.io/research/pytorch-extension-with-makefile/

ii) https://github.com/pytorch/extension-cpp

iii) https://github.com/huangtinglin/PyTorch-extension-Convolution - pytorch C++ extension for custom convolution

b) Some of the errors:

i) https://stackoverflow.com/questions/56470424/nvcc-missing-when-installing-cudatoolkit - missing nvcc in cudatoolkit

ii) https://github.com/HawkAaron/warp-transducer/issues/15 - cuda_runtime_api.h not found

c) Nvidia NGC container:

i) https://ngc.nvidia.com/catalog/containers/nvidia:pytorch

d) Setting up conda environment:

i) https://github.com/ollewelin/Installing-and-Test-PyTorch-C-API-on-Ubuntu-with-GPU-enabled - guide for CUDA 10.2 version pytorch

ii) https://pytorch.org/get-started/locally/ - Libtorch

iii) https://towardsdatascience.com/managing-cuda-dependencies-with-conda-89c5d817e7e1 - nvcc_linux-64 pkg

iv) https://developer.nvidia.com/cuda-10.2-download-archive?target_os=Linux&target_arch=x86_64&target_distro=Ubuntu&target_version=1804&target_type=runfilelocal - cudatoolkit - 10.2.89