Running ANN jobs in UVA-Rivanna

***** Make sure that Prof. Keller has added you to both the spin and spinquest groups in Rivanna. *****

Using Tensorflow with BKM2002-Formulation

1. Copy the sample files from the following Rivanna folder "/project/ptgroup/ANN_scripts/BKM-Formulation-Test/BKM-tf"
   cd /project/ptgroup/ANN_scripts/BKM-Formulation-Test/BKM-tf

2. Run the following commands on your terminal
   module load anaconda/2020.11-py3.8
   module load singularity/3.7.1
   module load tensorflow/2.1.0-py37
   following step is needed to run only once (it will copy the relevant .sif file to your /home directory)
   cp $$CONTAINERDIR/tensorflow-2.1.0-py37.sif /home/$USER

(make sure that you have the same module loads included in your grid.slurm file)

3. Run the following command to submit the job
   ./jobscript.sh <Name_of_Job> <Number_of_Replicas>

example:
./jobscript.sh CFF_BKM_tf_Test 10

Using PyTorch with BKM2002-Formulation

1. Copy the sample files from the following Rivanna folder "/project/ptgroup/ANN_scripts/BKM-Formulation-Test/BKM-PyTorch"
   cd /project/ptgroup/ANN_scripts/BKM-Formulation-Test/BKM-PyTorch

2. Run the following commands on your terminal
   module load anaconda/2020.11-py3.8
   module load singularity/3.7.1
   module load pytorch/1.8.1
   following step is needed to run only once (it will copy the relevant .sif file to your /home directory)
   cp $$CONTAINERDIR/pytorch-1.8.1.sif /home/$USER

(make sure that you have the same module loads included in your grid.slurm file)

3. Run the following command to submit the job
   $ ./jobscript.sh <Name_of_Job> <Number_of_Replicas>

example: $ ./jobscript.sh CFF_BKM_PyTorch_Test 10

Note:
If you download the code from GitHub to a Windows machine and then if you upload those files to Rivanna; then you will need to do the following steps

$ chmod u+x jobscript.sh
$ sed -i -e 's/\r$//' jobscript.sh
and
$ sed -i -e 's/\r$//' <all_files> in order to avoid any dos < - > unix conversions

** If you copy the fiels from /project/ptgroup/ANN_scripts/BKM-Formulation-Test/BKM-PyTorch then you don't have to do these above modification steps **

For more details check Zulkaida's folder on the github page: https://github.com/extraction-tools/ANN/tree/master/Zulkaida/BKM

Using Tensorflow with VA-Formulation

The following steps are for an example to submit a job for neural-net fit to 'N' number of kinematic settings in the data set (where N is an integer reflects to the range of kinematic settings which you will input in the sbatch command to submit the job).

1. Make sure that Prof. Keller has added you to both the spin and spinquest groups in Rivanna.
2. Copy the sample files from the following Rivanna folder "/project/ptgroup/ANN_scripts/VA-Formulation-Initial-Test"
   $ cd /project/ptgroup/ANN_scripts/VA-Formulation-Initial-Test

Here are the list of file that you need to have in your work directory:

Definitions
BHDVCStf.py
Lorentz_Vector.py
TVA1_UU.py
Data file dvcs_xs_May-2021_342_sets.csv
Main file Full_ML_fit_evaluation_Set2.py
Job submission file Job.slurm

3. Change the path(s) in the following files
   3.1) Highlighted line in "Job.slurm" file (please see below) with the correct path of 'your files'

```
bash-4.2$cat Job2.slurm
#!/usr/bin/env bash
#SBATCH -p standard
#SBATCH --output=result_%a.out
#SBATCH -c 1
#SBATCH -t 16:30:00
#SBATCH -A spinquest

module purge
module load anaconda/2020.11-py3.8
module load singularity/3.7.1
module load tensorflow/2.1.0-py37

singularity run --nv /home/$USER/tensorflow-2.1.0-py37.sif /home/cee9hc/ANN_GPD_Calc_Test/Full_ML_fit_evaluation_Set2.py ${SLURM_ARRAY_TASK_ID}
```

   3.2) Similarly update the paths on "Full_ML_fit_evaluation_Set2.py" file
   Line numbers 22, 31, 154

4. For a quick test, you can change the "number of samples" to a small number to test (in other words "number of replicas") which is in line number 115: 'numSamples = 10' as an example. You can change this numSamples value to any number of replicas that you need.

5. Run the following commands on your terminal
   $ module load anaconda/2020.11-py3.8
   $ module load singularity/3.7.1
   $ module load tensorflow/2.1.0-py37
   $ cp $CONTAINERDIR/tensorflow-2.1.0-py37.sif /home/$USER
(make sure that you have the same module loads included in your Job.slurm file)

6. Run the following command
   $ sbatch --array=0-2 Job.slurm
Note: Here 0-14 means the number of kinematic settings that you want to run in parallel (this is parallelization of local fits), and as a part of the output you will see Results#.csv (where # is an integer number) files which contain distributions of Compton Form Factors (CFFs) from each (individual) local fit.

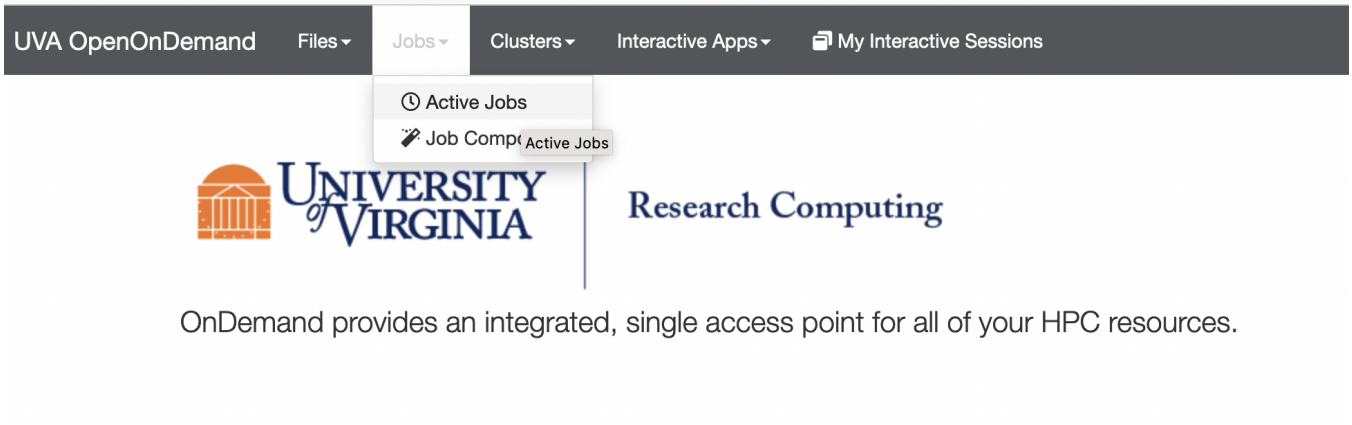
Below is an example of the above steps (up to step #6):

```
bash-4.2$mkdir ANN_test_code
bash-4.2$cd ANN_test_code/
bash-4.2$cp -r /project/ptgroup/ANN_scripts/Rivanna_test_code_for_ANN/* .
bash-4.2$ls
BHDVCStf.py  Full_ML_fit_evaluation_Set2.py  Job.slurm  Lorentz_Vector.py  TVA1_UU.py  dvcs_xs_May-2021_342_sets.csv  readme.txt
bash-4.2$pwd
/home/cee9hc/ANN_test_code
bash-4.2$vim Full_ML_fit_evaluation_Set2.py
bash-4.2$vim Job.slurm
bash-4.2$module load anaconda/2020.11-py3.8
bash-4.2$ module load singularity/3.7.1
To execute the default application inside the container, run:
singularity run --nv $CONTAINERDIR/tensorflow-2.1.0-py37.sif

bash-4.2$ module load tensorflow/2.1.0-py37
To execute the default application inside the container, run:
singularity run --nv $CONTAINERDIR/tensorflow-2.1.0-py37.sif

bash-4.2$ cp $CONTAINERDIR/tensorflow-2.1.0-py37.sif /home/$USER
```

7. After you submit your job:
   * You can view your jobs using the web-browser (please see the following screen-shots)



   * You can find commands to check the status of your job, cancel job(s), other commands related to handling jobs using .slurm file etc. using the following page
     https://www.rc.virginia.edu/userinfo/rivanna/slurm/

8. At the end of your job, you will find several types of output files (please see the the following screenshot)
   Results*.csv These files contain CFFs distributions corresponding to each kinematic setting
   best-netowrk*.hdf5 These files are the 'best'/'optimum' neural-network files for each kinematic setting
   result_*.out These files contain the output while it's been running for each kinematic setting

```
bash-4.2$ls
BHDVCStf.py                    Lorentz_Vector.py  Results2.csv  best-network0.hdf5  dvcs_xs_May-2021_342_sets.csv  result_1.out
Full_ML_fit_evaluation_Set2.py Results0.csv       TVA1_UU.py    best-network1.hdf5  readme.txt                     result_2.out
Job.slurm                      Results1.csv       __pycache__   best-network2.hdf5  result_0.out
bash-4.2$
```

Note: The true CFFs values which were used to generate these pseudo-data are given in 'https://github.com/extraction-tools/ANN/blob/master/Liliet/PseudoData2/dvcs_xs_May-2021_342_sets_with_trueCFFs.csv' **only** for the purpose of your comparison with what you obtain from your neural-net.

Important: Please consider that this is an example for running a neural-net fitting job on Rivanna for your reference.