

Training and Testing (On Google Colab)

Video explanation link for training and testing setup: -

https://drive.google.com/file/d/1XCvwjKsHp-PYli9JsQMQ-sO4sPvdcRK_/view?usp=sharing

Project GitHub repository link: - <https://github.com/Wenxuan-1119/TransBTS>

Step 1: - Download the Training dataset from this link: -

https://drive.google.com/drive/folders/1wrgQote5E2FzyCJ8SaNH_d_DAh2jKULd?usp=sharing

Upload these files in a folder on your drive and name it 'MICCAI_BraTS2020_TrainingData'

Step 2: - Download the Validation dataset from this link: -

<https://drive.google.com/drive/folders/1EZ8UXxgZT9eSHqMa6eINP1ghvdU19sLn?usp=sharing>

Upload these files in a folder on your drive and name it 'MICCAI_BraTS2020_ValidationData'

Step 3: - Download the training and testing files from this link: -

https://drive.google.com/drive/folders/1cJ_8CzGzZKMfsyrSqIJ3PY0-ecEAkau8?usp=sharing

Upload these files in a folder on your drive and name it 'TransBTS-main'

Step 4: - Open a new Google Colab notebook and link it to your drive.

Step 5: - Copy and paste the code from below which will download the necessary files and libraries for training and testing.

```
#####
```

```
!pip uninstall torch
```

```
!pip install torch==1.6.0 torchvision
```

```
!pip install setproctitle
```

```
!pip install tensorboardx
```

```
!pip install pickle5
```

#go to /usr/local/lib/python3.7/dist-packages/pandas/io/pickle.py and change the first line as "import pickle5 as pickle" instead of just "import pickle"

#####

After this, run the cell and make sure you edit the pickle file as mentioned in the last line of the above code. Press the CTRL key and hover over the directory of the pickle file and the cursor will change to a hand with the index finger out. Then right click and the file will open and then edit the first line as mentioned above.

Step 6: - Run the below code in another cell and the training will start.

```
!python -m torch.distributed.launch --nproc_per_node=1
/content/drive/MyDrive/TransBTS-main/train.py --resume_dir
'/content/drive/MyDrive/TransBTS-main/checkpoint/TransBTS2022-03-29/model_epoch_935.pth'
--load=True --save_freq=1 --num_workers=2 --batch_size=4 --train_dir
'/content/drive/MyDrive/MICCAI_BraTS2020_TrainingData' --valid_dir
'/content/drive/MyDrive/MICCAI_BraTS2020_ValidationData' --train_file
'/content/drive/MyDrive/TransBTS-main/train.txt' --valid_file
'/content/drive/MyDrive/TransBTS-main/valid.txt'
```

*** Remove --resume_dir and --load arguments if you are gonna start training from the first epoch! ***

Step 7: - After the training is completed the models trained will be put in the folder named 'checkpoint' inside the TransBTS main folder. Also logs for every epoch you train can be found in a .txt file in the 'log' folder of TransBTS main folder.

Step 8: - We use the validation dataset for testing since the dataset does not have a separate testing dataset as this dataset is a part of an international level competition. So, after training is over and you train a model which now lies in the checkpoint folder, you can now test it on the validation dataset. Run the below code for testing.

```
!python /content/drive/MyDrive/TransBTS-main/test.py --snapshot=False --post_process=False
--load_file
'/content/drive/MyDrive/TransBTS-main/checkpoint/TransBTS2022-03-29/model_epoch_935.pth'
--num_workers=2 --output_dir='/content/drive/MyDrive/' --valid_dir
'/content/drive/MyDrive/MICCAI_BraTS2020_ValidationData' --valid_file
'/content/drive/MyDrive/TransBTS-main/valid.txt'
```

*** Make sure you put the correct directory in place of the '--load_file' argument which is the directory to the trained model in the 'checkpoint' folder! ***

Step 9: - If you make the `--snapshot` argument to `True` you can see the outputs of the model in a folder called 'visualization' which will now be created in your Google Drive automatically. Also, there would be another folder named 'submission'. After the testing is over the files from the submission can now be uploaded to the UPENN website mentioned above to get the dice scores.

.....
IMPORTANT NOTES
.....

1. The original link of the dataset can be found from here: - <https://ipp.cbica.upenn.edu/>
2. We downloaded the dataset from the above link itself after creating an account and filling out all the details for the TransBTS 2020 dataset. It may take 2-3 days to get the approval.
3. The links KiTS and the LiTS dataset can be found here:
KiTS19: - <https://github.com/neheller/kits19>
LiTS17: - <https://competitions.codalab.org/competitions/17094>
4. We have not uploaded the datasets and the trained models for the KiTS and LiTS on our Google Drive unlike the Brats2020 because of memory limitations on our drive. So, for training and testing on these datasets you would need to download them explicitly and compress the datasets and then upload them.
5. Each of the datasets combined training and validation are at least up to 70 Gigs or more. Hence to upload them on our drive or other platforms we had to use a lossless compression algorithm since there is not enough space and bandwidth to upload them in pickle format. (More explanation in the video).
6. We mostly trained on the Agave cluster platform during which we had to create our own anaconda environment and then run the train and test files. The code required is the same for training and testing except you would be required to change the directories of datasets to access them accordingly. (More explanation in the video).
7. The files pertaining to the model code are as follows: `IntmdSequential.py`, `PositionalEncoding.py`, `TransBTS_downsample8x_skipconnection.py`, `Transformer.py`, and `Unet_skipconnection.py`

.....
Code snippet for converting data from pickle to bz2 compression
.....

Step 1: - Convert the `.nii.gz` files to `.pkl` files. This step has actually been mentioned on the GitHub repository itself. You need to run the `preprocess.py` file from the TransBTS main folder. Please make sure to change the 'root' argument directories which point to the respective dataset folder as mentioned in the `preprocess.py` file. You can also use this file for the LiTS, KiTS datasets. However, in case of the LiTS dataset the files first need to be unzipped since the extension of these files is `nii.zip` and then use the `gzip` library to convert it to `nii.gz` after which you can pass it to the `preprocess.py`. For KiTS the dataset is

already in .nii.gz format. (.nii.gz format is multi-dimensional neuroimaging data format. In other words, the MRI is usually stored in this format)

Step 2: - Run the data_compression.py from the 'Codes' folder in the given package. This will convert the .pkl files to .bz2 meaning that the compression has been applied. Again, make sure that the directories are changed according to where the dataset lies and where you would like the code to dump the new compressed files.

****** It should be noted that the compression is done only to transfer the data between multiple platforms such as local machine, Google Colab, Agave Cluster (ASU specific). It does not have any effect on the performance during training or testing. Also, it does not have any affiliation with the batch size during training. *****

***** Annotated codes can be found in the 'Codes' folder. In the train_test_codes_on_colab.py the same codes mentioned above for installing libraries, training, and testing on Google Colab can be found except they are annotated. *****

.....
Code snippet saving intermediary images
.....

Step 1: Replace TransBTS_downsample8x_skipconnection.py in the TransBTS-main/models/ directory with our annotated version.

Step 2: Run the train file normally and images will be saved within the directory

***** To edit the images being displayed go to line 229. This is the block of code that manages the creation of images. Here one can change the slice of the image being saved as well as to display the encoding or decoded images. *****