Status report on

DNNROI sigproc & wirecell-dnn

Hokyeong Nam
Chung-Ang University

Outline

- Sigproc. w/ dnnroi
  - Solved issue
  - Execution on the dunegpvm and the WC cluster

- “wirecell-dnn” validation
  - Timing performance compared to train3.py
  - Issues in loss trend, trained models

- Summary & Plan

DNNROI solved issue: large memory usage

```
channels_break_roi_1st0  Dataset {2560}
channels_break_roi_2nd0  Dataset {2560}
channels_cleanup_roi0    Dataset {2560}
channels_decon_charge0   Dataset {2560}
channels_dnnsp0          Dataset {2560}
channels_extend_roi0     Dataset {2560}
channels_gauss0          Dataset {2560}
channels_loose_lf0       Dataset {2560}
channels_mp2_roi0        Dataset {2560}
channels_mp3_roi0        Dataset {2560}
channels_shrink_roi0     Dataset {2560}
channels_tight_lf0       Dataset {2560}
frame_break_roi_1st0     Dataset {6000, 2560}
frame_break_roi_2nd0     Dataset {6000, 2560}
frame_cleanup_roi0       Dataset {6000, 2560}
frame_decon_charge0      Dataset {6000, 2560}
frame_dnnsp0             Dataset {6000, 2560}
frame_extend_roi0        Dataset {6000, 2560}
frame_gauss0             Dataset {6000, 2560}
frame_loose_lf0          Dataset {6000, 2560}
frame_mp2_roi0           Dataset {6000, 2560}
frame_mp3_roi0           Dataset {6000, 2560}
frame_shrink_roi0        Dataset {6000, 2560}
frame_tight_lf0          Dataset {6000, 2560}
```

❖ Old files for the ML training

```
channels_break_roi_1st0  Dataset {3027}
channels_break_roi_2nd0  Dataset {3022}
channels_cleanup_roi0    Dataset {3023}
channels_decon_charge0   Dataset {582997}
channels_extend_roi0     Dataset {1266}
channels_gauss0          Dataset {2548}
channels_loose_lf0       Dataset {593611}
channels_mp2_roi0        Dataset {3864}
channels_mp3_roi0        Dataset {771}
channels_shrink_roi0     Dataset {3022}
channels_tight_lf0       Dataset {792153}
frame_break_roi_1st0     Dataset {3027, 5978}
frame_break_roi_2nd0     Dataset {3022, 5978}
frame_cleanup_roi0       Dataset {3023, 5978}
frame_decon_charge0      Dataset {582997, 6000}
frame_extend_roi0        Dataset {1266, 5734}
frame_gauss0             Dataset {2548, 5732}
frame_loose_lf0          Dataset {593611, 6000}
frame_mp2_roi0           Dataset {3864, 2785}
frame_mp3_roi0           Dataset {771, 2785}
frame_shrink_roi0        Dataset {3022, 5978}
frame_tight_lf0          Dataset {792153, 6000}
```

❖ Generated bad files

```
channels_break_roi_1st0  Dataset {2560}
channels_break_roi_2nd0  Dataset {2560}
channels_cleanup_roi0    Dataset {2560}
channels_decon_charge0   Dataset {2560}
channels_extend_roi0     Dataset {2560}
channels_gauss0          Dataset {2560}
channels_loose_lf0       Dataset {2560}
channels_mp2_roi0        Dataset {2560}
channels_mp3_roi0        Dataset {2560}
channels_shrink_roi0     Dataset {2560}
channels_tight_lf0       Dataset {2560}
frame_break_roi_1st0     Dataset {2560, 6000}
frame_break_roi_2nd0     Dataset {2560, 6000}
frame_cleanup_roi0       Dataset {2560, 6000}
frame_decon_charge0      Dataset {2560, 6000}
frame_extend_roi0        Dataset {2560, 6000}
frame_gauss0             Dataset {2560, 6000}
frame_loose_lf0          Dataset {2560, 6000}
frame_mp2_roi0           Dataset {2560, 6000}
frame_mp3_roi0           Dataset {2560, 6000}
frame_shrink_roi0        Dataset {2560, 6000}
frame_tight_lf0          Dataset {2560, 6000}
```

❖ Good files after debugging

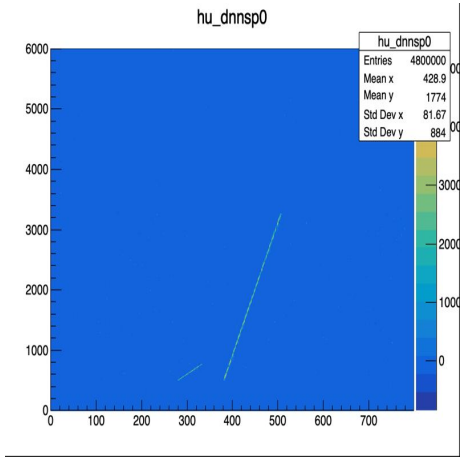
Different ordering is convention by Haiwang and Sergey

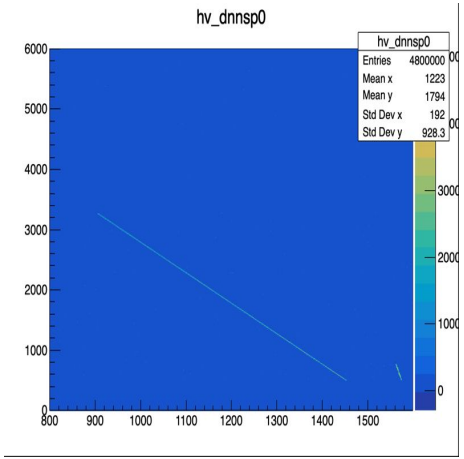
- Sigproc w/ dnnroi (wct-sim-drift-deposplat.jsonnet) raised a memory problem
  - MEM ~40 GB, the job is killed by system
- Succeed to run on the WC cluster, the output file showed wrong wire channel & time tick
- Debugged with Haiwang and fixed the issue in **unique trace** (https://github.com/WireCell/wire-cell-toolkit/commit/83ac1e8070289b80daa68619a1aaa47cbc03bf0e)

DNNROI SigProc on dunegpvm

```
[01:15:08.934] I [ timer  ] Timer: WireCell::Pytorch::DNNROIFinding : 10.77 sec
[01:15:08.934] I [ timer  ] Timer: WireCell::Pytorch::DNNROIFinding : 9.66 sec
[01:15:08.934] I [ timer  ] Timer: WireCell::SigProc::OmnibusSigProc : 9.17 sec
```

```
CPU: 100.0%, MEM: 24.3%
CPU: 94.1%, MEM: 29.9%
CPU: 94.1%, MEM: 33.5%
CPU: 94.1%, MEM: 33.5%
```



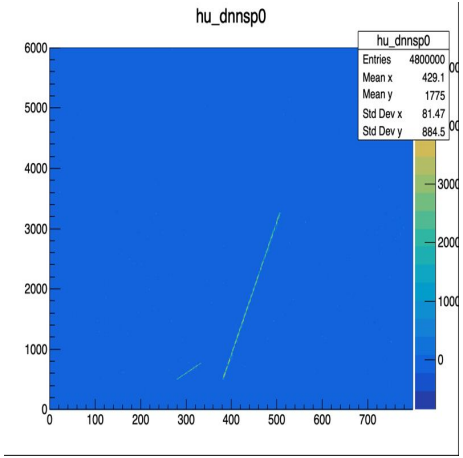


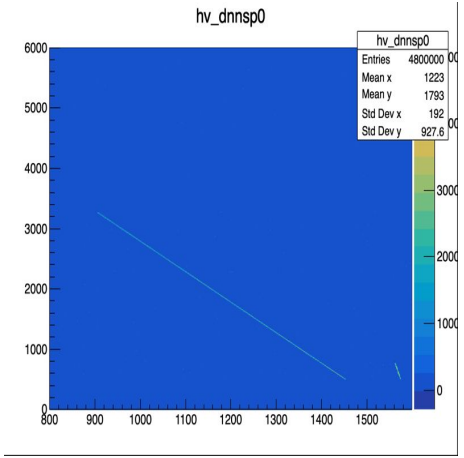
- DNNROI takes twice the time
  - Traditional ROI finding: 9.17s
  - DNNROI finding: 20.43s
- Peak memory usage (entire process): 3.92/11.7 GB ≈ 33.5%

DNNROI SigProc on the WC cluster

```
[02:20:26.575] I [ timer  ] Timer: WireCell::SigProc::OmnibusSigProc : 5.82 sec
[02:20:26.575] I [ timer  ] Timer: WireCell::Pytorch::DNNROIFinding : 5.71 sec
[02:20:26.575] I [ timer  ] Timer: WireCell::Pytorch::DNNROIFinding : 5.44 sec
```

```
CPU: 94.1%, MEM: 1.1%
CPU: 94.1%, MEM: 0.9%
CPU: 100.0%, MEM: 1.5%
CPU: 100.0%, MEM: 1.5%
```





- DNNROI takes twice the time
  - Traditional ROI finding: 5.82s
  - DNNROI finding: 11.15s
- Peak memory usage (entire process): 3.86/257.2 GB ≈ 1.5%

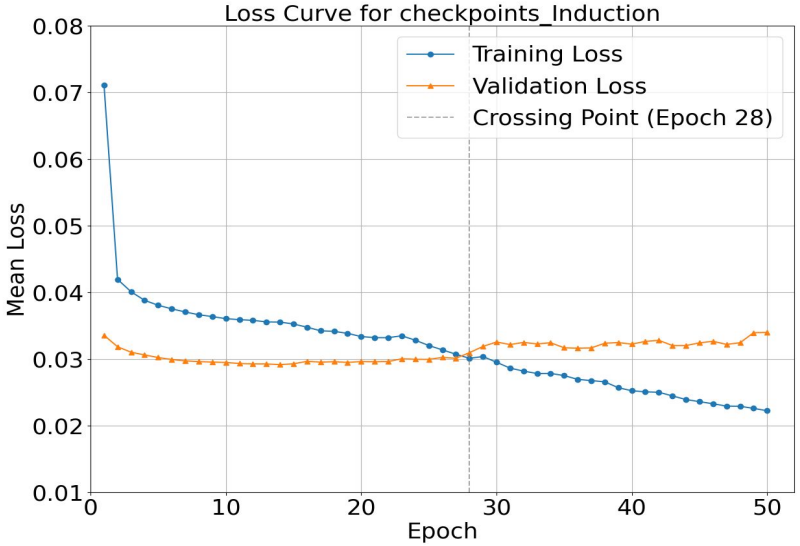
wirecell-dnn setup & training

```
Create dataset in 2.135e+02 s
Total training time: 764.51 seconds
```

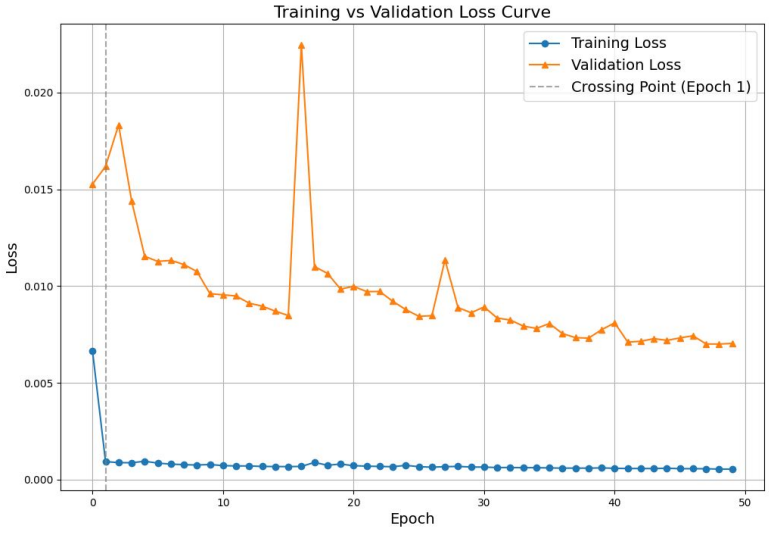
- Training UNet with wirecell-dnn and train3.py on the WC cluster w/ full (118) hdf5 files
  - Total elapsed time (wirecell-dnn): **18m 01s**
  - Total elapsed time (train3.py): **25m 19s**

- Dataset: /nfs/data/1/hnam/train_data_PDHD_fixedbug_separateWC
- Epochs: 50
- Batch Size: 10
- Train Ratio: 0.9, Validation Ratio: 0.1
- Learning Rate: 0.1

Issues: Train vs Val loss curve



❖ mdeol from train3.py



❖ mdeol from wirecell-dnn train

Issues: .pth file structure diff.

```
>>> print(data.keys())
odict_keys(['inc.conv.conv.0.weight', 'inc.conv.conv.0.bias', 'inc.conv.conv.1.w
eight', 'inc.conv.conv.1.bias', 'inc.conv.conv.1.running_mean', 'inc.conv.conv.1
.running_var', 'inc.conv.conv.1.num_batches_tracked', 'inc.conv.conv.3.weight',
'inc.conv.conv.3.bias', 'inc.conv.conv.4.weight', 'inc.conv.conv.4.bias', 'inc.c
onv.conv.4.running_mean', 'inc.conv.conv.4.running_var', 'inc.conv.conv.4.num_ba
tches_tracked', 'down1.mpconv.1.conv.0.weight', 'down1.mpconv.1.conv.0.bias', 'd
own1.mpconv.1.conv.1.weight', 'down1.mpconv.1.conv.1.bias', 'down1.mpconv.1.conv
.1.running_mean', 'down1.mpconv.1.conv.1.running_var', 'down1.mpconv.1.conv.1.nu
m_batches_tracked', 'down1.mpconv.1.conv.3.weight', 'down1.mpconv.1.conv.3.bias'
, 'down1.mpconv.1.conv.4.weight', 'down1.mpconv.1.conv.4.bias', 'down1.mpconv.1.
conv.4.running_mean', 'down1.mpconv.1.conv.4.running_var', 'down1.mpconv.1.conv.
4.num_batches_tracked', 'down2.mpconv.1.conv.0.weight', 'down2.mpconv.1.conv.0.b
ias', 'down2.mpconv.1.conv.1.weight', 'down2.mpconv.1.conv.1.bias', 'down2.mpcon
v.1.conv.1.running_mean', 'down2.mpconv.1.conv.1.running_var', 'down2.mpconv.1.c
```

```
>>> print(data.keys())
dict_keys(['runs', 'epochs', 'model_state_dict', 'optimizer_state_dict'])
>>> print(data["model_state_dict"].keys())
odict_keys(['unet.down_dconv_0.0.weight', 'unet.down_dconv_0.0.bias', 'unet.down
_dconv_0.1.weight', 'unet.down_dconv_0.1.bias', 'unet.down_dconv_0.1.running_mea
n', 'unet.down_dconv_0.1.running_var', 'unet.down_dconv_0.1.num_batches_tracked'
, 'unet.down_dconv_0.3.weight', 'unet.down_dconv_0.3.bias', 'unet.down_dconv_0.4
.weight', 'unet.down_dconv_0.4.bias', 'unet.down_dconv_0.4.running_mean', 'unet.
down_dconv_0.4.running_var', 'unet.down_dconv_0.4.num_batches_tracked', 'unet.do
wn_dconv_1.0.weight', 'unet.down_dconv_1.0.bias', 'unet.down_dconv_1.1.weight',
'unet.down_dconv_1.1.bias', 'unet.down_dconv_1.1.running_mean', 'unet.down_dconv
_1.1.running_var', 'unet.down_dconv_1.1.num_batches_tracked', 'unet.down_dconv_1
.3.weight', 'unet.down_dconv_1.3.bias', 'unet.down_dconv_1.4.weight', 'unet.down
```

- The model trained with wirecell-dnn has more info.
- Naming convention and structure differences → revising to-ts.py

Summary & Plan

- Solved large memory issues in sigproc w/ dnnroi
- Checked time consumption for traditional ROI finding and DNNROI
- Tried to train the UNet with wirecell-dnn and found diff. in loss trend
- Next steps for dnnroi sigproc.
  - Check 1D wave from of the output
  - Run with PDHD data
  - Find a way to reduce time consumption
  - Measure the computing resources and time used during the inference stage more specifically
- Next steps for wirecell-dnn validation
  - Check chunked training & data load works as expected
  - Confirm train3.py and wirecell-dnn importing the same UNet as same structure