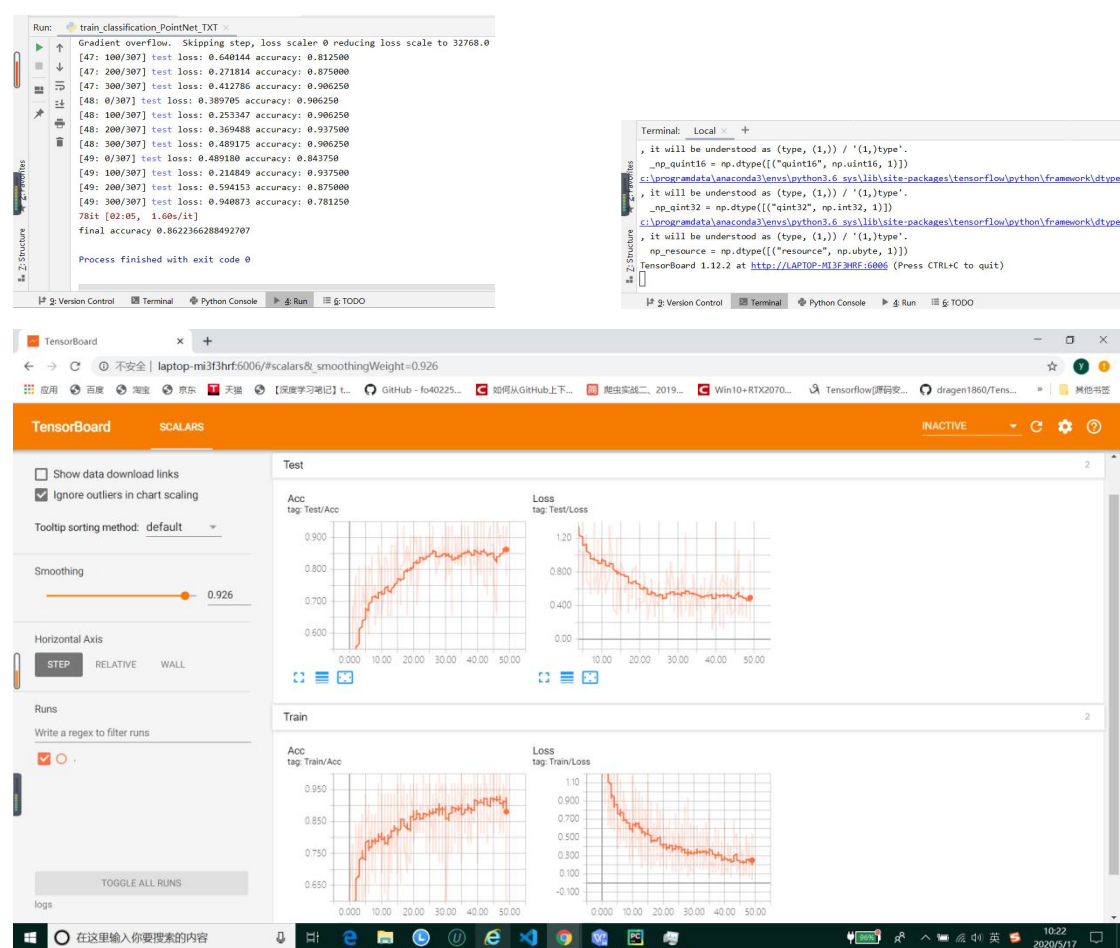


在 GitHub 项目 PointNet example: <https://github.com/fxia22/pointnet.pytorch>, 上基础改写, 分别实现 PoingNet 和 VoxelFeatureEncoding 两种模型并做了对比, 数据集划分按官方 txt 格式 ModelNet40 Dataset, 数据处理对点云的 X 和 Z 轴做旋转的数据增广。另外, 使用了 tensorboardX 库对模型在训练和测试过程中的损失值和分类精度记录, 而且为了加快模型训练时间, 使用 apex 库进行半精度训练模式。实验环境为 win10 下 RTX2060 6G 显存, PyTorch 深度学习框架。两个模型实验设置参数 batchSize=32, nepoch=50, num_points=1024 均基本相同, 最后的分类精度分别约为 86.22%和 85.82%。理论上应该 VoxelFeatureEncoding 比 PointNet 测试情况好, 但我是串联两个 PointNet 去实现实现 VoxelFeatureEncoding 时, 中间主要是删除了 feature_transform 设置, 去掉这部分 loss 值的学习, 有待继续调试模型结构和相关参数设置。

1、PoingNet 模型的训练/测试 损失值、精度曲线



2、VoxelFeatureEncoding 模型的训练/测试 损失值、精度曲线

```
Run: train_classification_VoxelfeatureEncoding_T...
[48: 0/307] test loss: 0.485911 accuracy: 0.906250
Gradient overflow. Skipping step, loss scalar 0. Reducing loss scale to 32768.0
[48: 100/307] test loss: 0.521697 accuracy: 0.781250
[48: 200/307] test loss: 1.148342 accuracy: 0.687500
[48: 300/307] test loss: 0.430817 accuracy: 0.843750
[49: 0/307] test loss: 0.527783 accuracy: 0.906250
[49: 100/307] test loss: 0.406210 accuracy: 0.812500
[49: 200/307] test loss: 0.339990 accuracy: 0.906250
[49: 300/307] test loss: 0.069489 accuracy: 1.000000
78it [02:03, 1.59s/it]
final accuracy 0.8581847649918962
Process finished with exit code 0
```

```
Terminal: Local
, it will be understood as (type, (1,)) / '(1,)type'.
_np_quint16 = np.dtype([('quint16', np.uint16, 1)])
c:\programdata\anaconda3\envs\python3.6_sys\lib\site-packages\tensorflow\python\framework\dtypes.py:527:
, it will be understood as (type, (1,)) / '(1,)type'.
_np_quint32 = np.dtype([('quint32', np.int32, 1)])
c:\programdata\anaconda3\envs\python3.6_sys\lib\site-packages\tensorflow\python\framework\dtypes.py:532:
, it will be understood as (type, (1,)) / '(1,)type'.
_np_resource = np.dtype([('resource', np.ubyte, 1)])
TensorBoard 1.12.2 at http://LAPTOP-MI3F3MRF-6006 (Press CTRL+C to quit)
```

