unet3D tutorial

This tutorial illustrates how to use unet3D.py to highlight cell contours. We will use some synthetic data generated with *generateSynthCellsImage3D* from *gpfunctions.py.*

The script unet3D.py is very similar to unet2D.py in terms of parameters – the only difference is that unet3D.py only works on single-channel images, thus the parameter *nChannels* is not present. Given the similarities, we recommend the reader to first look at the unet2D tutorial.

Note that this tutorial merely intends to show how to use the script, not how to get the best possible segmentation results.

1. Data Generation

The auxiliary notebook *t04_unet3D.ipynb* contains 3 code cells to generate training, test, and deployment data. Please refer to that file for this step.

2. Training

With the data generated as above, the following configuration could be used when training from scratch:

```
restoreVariables = False
train = True
test = False
deploy = False
deployImagePathIn = 'tutorials/DataForUNet3D/Deploy_In/I00000_Img.tif'
deployFolderPathIn = 'tutorials/DataForUNet3D/Deploy_In'
deployFolderPathOut = 'tutorials/DataForUNet3D/Deploy_Out'
imSize = 60
nClasses = 2
batchSize = 4
modelPathIn = 'Models/unet3D_v0.ckpt'
modelPathOut ='Models/unet3D_v0.ckpt'
reSplitTrainSet = True
trainSetSplitPath = 'Models/trainSetSplit3D.data'
logDir = 'Logs/unet3D'
logPath = 'Logs/unet3D_TestSample.tif'
imPath = 'tutorials/DataForUNet3D/Train_60'
imPathTest = 'tutorials/DataForUNet3D/Test'
nFeatMapsList = [16,32,64]
learningRate = 0.00001
nEpochs = 50
```

```
useGPU = True
```

3. Fine-Tuning

To fine-tune a previously trained model, one could use the following settings. Notice the differences from the configuration above are in *restoreVariables*, *modelPathOut*, and *reSplitTrainSet*.

```
restoreVariables = True
train = True
test = False
deploy = False
deployImagePathIn = 'tutorials/DataForUNet3D/Deploy_In/I00000_Img.tif'
deployFolderPathIn = 'tutorials/DataForUNet3D/Deploy_In'
deployFolderPathOut = 'tutorials/DataForUNet3D/Deploy_Out'
imSize = 60
nClasses = 2
batchSize = 4
modelPathIn = 'Models/unet3D_v0.ckpt'
modelPathOut ='Models/unet3D_v1.ckpt'
reSplitTrainSet = False
trainSetSplitPath = 'Models/trainSetSplit3D.data'
logDir = 'Logs/unet3D'
logPath = 'Logs/unet3D_TestSample.tif'
imPath = 'tutorials/DataForUNet3D/Train_60'
imPathTest = 'tutorials/DataForUNet3D/Test'
nFeatMapsList = [16,32,64]
learningRate = 0.00001
nEpochs = 50
useGPU = True
```

4. Testing

In testing mode, the configuration is like so:

```
restoreVariables = True
train = False
test = True
deploy = False
deployImagePathIn = 'tutorials/DataForUNet3D/Deploy_In/I00000_Img.tif'
deployFolderPathIn = 'tutorials/DataForUNet3D/Deploy_In'
deployFolderPathOut = 'tutorials/DataForUNet3D/Deploy_Out'
imSize = 60
```

```
nClasses = 2
batchSize = 4
modelPathIn = 'Models/unet3D_v0.ckpt'
modelPathOut ='Models/unet3D_v1.ckpt'
reSplitTrainSet = False
trainSetSplitPath = 'Models/trainSetSplit3D.data'
logDir = 'Logs/unet3D'
logPath = 'Logs/unet3D_TestSample.tif'
imPath = 'tutorials/DataForUNet3D/Train_60'
imPathTest = 'tutorials/DataForUNet3D/Test'
nFeatMapsList = [16,32,64]
learningRate = 0.00001
nEpochs = 50
useGPU = True
```

5. Deployment

And here’s what deployment mode settings look like:

```
restoreVariables = True
train = False
test = False
deploy = True
deployImagePathIn = 'tutorials/DataForUNet3D/Deploy_In/I00000_Img.tif'
deployFolderPathIn = 'tutorials/DataForUNet3D/Deploy_In'
deployFolderPathOut = 'tutorials/DataForUNet3D/Deploy_Out'
imSize = 60
nClasses = 2
batchSize = 4
modelPathIn = 'Models/unet3D_v0.ckpt'
modelPathOut ='Models/unet3D_v1.ckpt'
reSplitTrainSet = False
trainSetSplitPath = 'Models/trainSetSplit3D.data'
logDir = 'Logs/unet3D'
logPath = 'Logs/unet3D_TestSample.tif'
imPath = 'tutorials/DataForUNet3D/Train_60'
imPathTest = 'tutorials/DataForUNet3D/Test'
nFeatMapsList = [16,32,64]
learningRate = 0.00001
nEpochs = 50
useGPU = True
```