1. Collect\_indoor2d\_data.py

Collect data from txt or numpy file, and make first preprocess

Used function: indoor3d\_util.collect\_point\_label

Convert original dataset files to data\_label file (each line is XYZRGBL)

points = np.loadtxt(f)

# points read from txt file  
labels = np.ones((points.shape[0],1)) \* g\_class2label[cls]

#label is produced through the file name.

data\_label will combime points and label as a row. So each ros will contain 7 elements.

data\_label (each line is XYZRGB L). The minimum of xyz has already been

subtracted hier.

Finally will be saved as a numpy fiel used for next step, gen\_indoor3d\_h5.py

2. gen\_indoor3d\_h5.py

Filelist all\_data\_label.txt be splited into data\_label\_filename(npy files name)

A list of all npy files made in last step will be transformed to h5 format. It’s should be made by ourself

output\_dir

batch\_data\_dim = [H5\_BATCH\_SIZE] + data\_dim [1000 4096 9]

batch\_label\_dim = [H5\_BATCH\_SIZE] + label\_dim [1000 4096]

room2blocks\_wrapper\_normalized

load data\_label\_filename and invoke room2blocks\_plus\_normalized

rgb will be normlaized into 0-1

data size [1136617 6]

call room2blocks

Prepare block training data. Divide room area into many blocks

And randomly subsample data to guarantee the points number in each block is 4096. sample\_data\_label

Got data\_batch [80 4096 6] 80 is the num of the blocks, 4096 is the pointsnum of each block

new\_data\_batch

6,7,8 = 0,1,2/max of xyz,

x,y - minx+blocksize/2,miny+blocksize/2

for each block centralize XYZ, add normalized XYZ as 678 channels.

insert\_batch