Tutorial-2: read and write SAR data using python

Dr. Subhadip Dey
Optical and SAR Sensing Laboratory (OptiSARLab), AGFE, IITKGP

1. Read binary SAR image (real valued channels) using python struct

```
import struct
import numpy as np
import matplotlib.pyplot as plt


fname = r'ENTER THE FULL FILE PATH HERE'
f = open(fname, "rb")

byte_gap = 4
init_byte = -4 # We keep it -4 because we want to start reading from 0
data = []

tot_byte = 81204 # Check the total size of file in bytes

while init_byte <= (tot_kb-8): # substract 8 because it reads 4 bytes
at once
    f.seek(init_byte+byte_gap)
    byte = f.read(4)
    data_pix = struct.unpack("f", byte)

    data.append(data_pix)

    init_byte = init_byte+byte_gap # We want to move cursor 4 byte and
then read 4 byte again

all_data = np.array(data)
all_data = all_data.reshape([201, 101])# Check the size of image from
hdr or from the config file

plt.imshow(all_data, cmap = 'gray')
```

2. Now, we can read the same using gdal

```
from osgeo import gdal
import numpy as np
import matplotlib.pyplot as plt


file = r'ENTER THE FULL FILE PATH HERE'
ds = gdal.Open(file) # Opens the file as dataset
band = ds.GetRasterBand(1) # Get band number 1
arr = band.ReadAsArray() # Read the band as an array

plt.imshow(arr, cmap = 'gray')
```

3. Write bin file using gdal

```
from osgeo import gdal
import numpy as np

# First read a reference data whose geometric features we will copy in
our newly obtained data
refData = r'ENTER THE FULL FILE PATH HERE'

# Generate a new data. In this case a 2D data is generated using numpy
wdata = np.random.random((201,101))

# Set the filename for the new data to be saved in .bin format
file = r'PATH\New_data.bin'

ds = gdal.Open(refData)
[cols, rows] = wdata.shape

driver = gdal.GetDriverByName("ENVI")
outdata = driver.Create(file, rows, cols, 1, gdal.GDT_Float32)
#sets same geotransform as reference data
outdata.SetGeoTransform(ds.GetGeoTransform())
#sets same projection as reference data
outdata.SetProjection(ds.GetProjection())

outdata.SetDescription(file)
outdata.GetRasterBand(1).WriteArray(wdata)
outdata.FlushCache()
```

4. Enhance image visualization

Copy the complete code from 1

Min-Max stretching

```
plt.imshow(all_data, vmin = float(all_data.min()), vmax =
float(all_data.max()), cmap = 'gray')
```

Mean±standard deviation x n stretching

For example, n = 2,

```
mean = np.mean(all_data)
std = np.std(all_data)
vmin = mean - std * 2
vmax = mean + std * 2
plt.imshow(all_data, vmin = vmin, vmax = vmax, cmap = 'gray')
```

Cumulative count cut (lower percent, upper percent)

```
low_frac = 0.02
upper_frac = 0.98

flat_data = all_data.ravel()
flat_data = flat_data[np.isfinite(flat_data)]
sorted_vals = np.sort(flat_data)
n = len(sorted_vals)
vmin = sorted_vals[int(low_frac * n)]
vmax = sorted_vals[int(upper_frac * n)]

plt.imshow(all_data, vmin = vmin, vmax = vmax, cmap = 'gray')
```