DATS 6203 Individual Final Project

Jinshun Su

This project investigates the effective utilization of the Autoencoder (AE) neural network model for online identification of the distribution system (DS) topology based on a large amount of real-time phasor measurement units (PMUs) detection data. The detail of this project is presented in our group report. My major contribution in this project is in data preprocessing section. I came up with a code for translated heatmap in Python from the data obtained by MATLAB Simulink. Other minor contributions from me in this project is that: i). discuss the possible way to improve model performance with my teammates who mainly work on building AE model; ii). discuss the reasonable study cases can be tested in our project; iii). serve as the main author in report writing and make slides for presentation. Next, I will present my individual work in data preprocessing section.

The total number of generated scenarios tested in our project is 19477. The PMU data is generated by simulating IEEE-34 node test feeder in MATLAB Simulink. The "parula" format obtained from MATLAB challenge us when we use this as input for our model. As shown in **Figure 1**, the white margin of the image are too wide. If it is used as input, it will cause the input data size to be too large and not compressible and affect the prediction accuracy on the neural network. So I use the pure data form .csv format of the PMU saved by MATLAB, and generate heatmaps in batches through python instead. Here, I used "seaborn.heatmap" library to generate the heatmap in Python [1]. At the beginning, I generated some general formats of heatmap (see **Figure 2**). Based on the discussion with my teammates, these heatmaps still have tiny issue – the color is so single that cannot distinguish the data boundary well. Later, I used a colormap format as "gist_rainbow" in generating image (see **Figure 3**) as the input of the our neural network model [2]. The snapshot of manuscript for generating heatmap is show in **Figure 4**. The original size of one heatmap is 497 by 371 pixels, we compressed it into 96 by 96 pixels in the input layer.

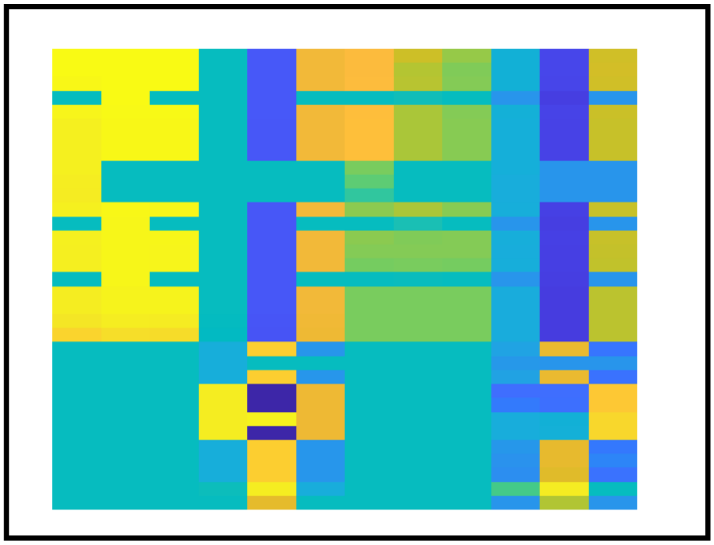

Figure 1: The "parula" Heatmap Obtained by MATLAB.

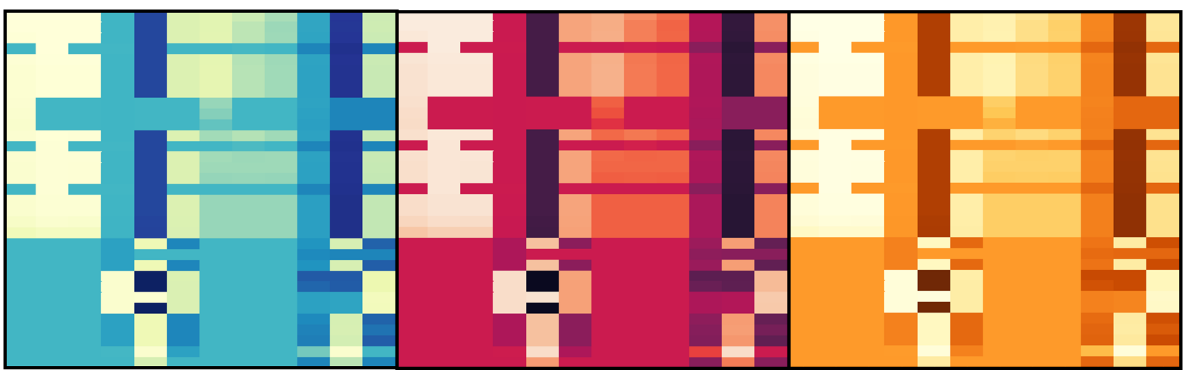

Figure 2: Several General Heatmap Formats.

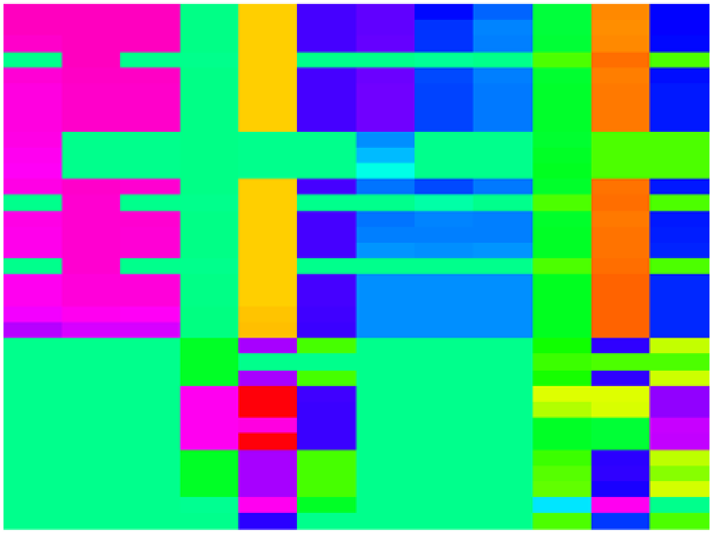

Figure 3: The "gist_rainbow" Heatmap Formats.

References

[1] "seaborn.heatmap." [Online] Available at: `https://seaborn.pydata.org/generated/seaborn.heatmap.html`, Accessed:2020.

[2] "How bad is your colormap?." [Online] Available at: `http://jakevdp.github.io/blog/2014/10/16/how-bad-is-your-colormap/`, Accessed:2020.

```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import os, cv2
from PIL import Image
import seaborn as sns
import tqdm

# https://seaborn.pydata.org/generated/seaborn.heatmap.html
# http://jakevdp.github.io/blog/2014/10/16/how-bad-is-your-colormap/

os.environ["CUDA_VISIBLE_DEVICES"] = "0"

def Heatmap_Generator(CATEGORIES, Path_From, Path_To,label):
    for label_i in label:
        for category in CATEGORIES:
            # print(category)
            data_path = os.path.join(Path_From+label_i, category)
            # print(data_path)
            save_path = os.path.join(Path_To+label_i, category)
            # print(save_path)
            try:
                os.mkdir(save_path)
            except OSError:
                print("Creation of the directory %s failed" % save_path)
            else:
                print("Successfully created the directory %s" % save_path)

            for csv in os.listdir(data_path):
                print('-')
                # print(csv)
                try:
                    # print(csv)
                    matrix_array = pd.read_csv(os.path.join(data_path, csv), header=None)
                    # print(os.path.join(data_path, csv))
                    img_list = matrix_array.values.tolist()
                    # print(np.asarray(img_list).shape)
                    img = sns.heatmap(img_list, vmin=-1.05, vmax=1.05, cmap="gist_rainbow", cbar=False)
                    img.axis('off')
                    img_name = os.path.join(save_path, csv.replace('.csv', '.png'))
                    # print(img_name)
                    plt.savefig(img_name, bbox_inches='tight', pad_inches=0.01)
                    # # plt.show()
                    # print('-')
                except Exception as e:  # in the interest in keeping the output clean...
                    pass


sns.set()
path = 'C:/Users/Administrator/Desktop/M2_Final/Data/'
print(path)
Path_From = path + '/Validation/AE/CNN_22PMU_MissingOneData_add10dB/'
Path_To = path + '/IMAGE/CNN_22PMU_MissingOneData_add10dB/'
CATEGORIES = ['1', '2', '3', '4', '5', '6', '7', '8']
label = ['Train/', 'Test/', 'Val/']
```

Figure 4: Snapshot of Manuscript in Python.