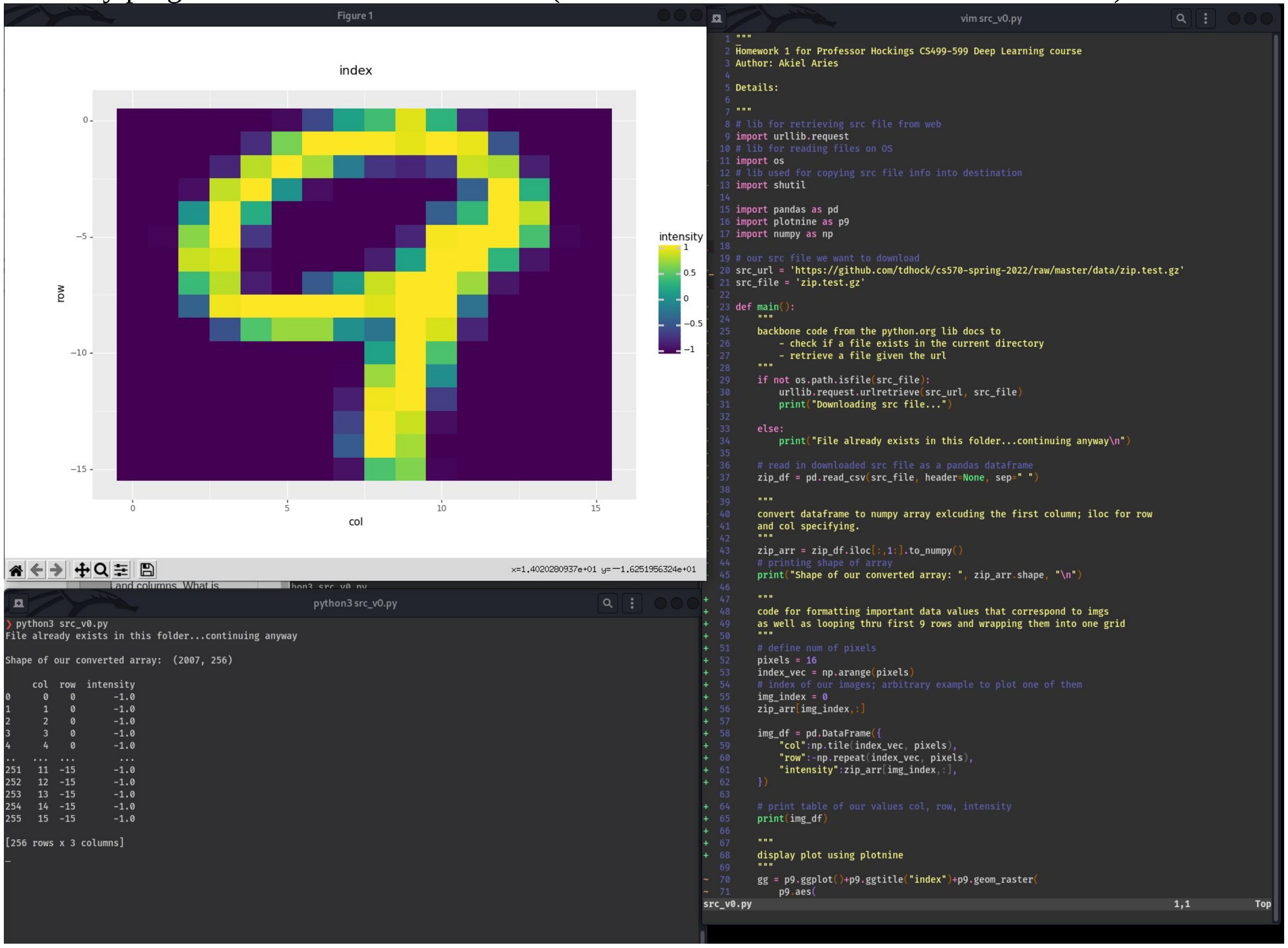
Akiel Aries
09/02/2022
CS499 – Deep Learning
Professor Hocking

Here is my program(I also attached the .py file to the assignment, I understand the deliverable in the assignment guidelines was a .pdf file with all requirements in it):

```
# lib for retrieving src file from web
import urllib.request
# lib for reading files on OS
import os

import shutil

import pandas as pd
import plotnine as p9
import numpy as np

# our src file we want to download
src_url = 'https://github.com/tdhock/cs570-spring-2022/raw/master/data/zip.test.gz'
src_file = 'zip.test.gz'

def main():
    """
    backbone code from the python.org lib docs to
        - check if a file exists in the current directory
        - retrieve a file given the url
    """
    if not os.path.isfile(src_file):
        urllib.request.urlretrieve(src_url, src_file)
        print("Downloading src file...")

    else:
        print("File already exists in this folder...continuing anyway\n")

    # read in downloaded src file as a pandas dataframe
    zip_df = pd.read_csv(src_file, header=None, sep=" ")

    """
    convert dataframe to numpy array exlcuding the first column; iloc for row
    and col specifying.
    """
    zip_arr = zip_df.iloc[:,1:].to_numpy()
    # printing shape of array
    print("Shape of our converted array: ", zip_arr.shape, "\n")
    """
    code for formatting important data values that correspond to imgs
    as well as looping thru first 9 rows and wrapping them into one grid
    """
    # define num of pixels
    pixels = 16
    index_vec = np.arange(pixels)
    # index of our images; arbitrary example to plot one of them
    img_index = 0
    zip_arr[img_index,:]

    img_df = pd.DataFrame({
        "col":np.tile(index_vec, pixels),
        "row":-np.repeat(index_vec, pixels),
        "intensity":zip_arr[img_index,:],
    })

    # print table of our values col, row, intensity
    print(img_df)

    """
    display plot using plotnine
    """
    gg = p9.ggplot()+p9.ggtitle("index")+p9.geom_raster(
        p9.aes(
            # assign vals to our img properties
            label = "0-9",
            x = "col",
            y = "row",
            fill = "intensity",),
        data = img_df) #+ p9.facet_grid("~col")

    """
    attempted to wrap our images 0-9 to be displayed together but was not sure what the
    observation and label variables are supposed to signify, I was thinking if I set
    the initial condition of the loop to img_index 0 and go up to 9, put them in a
    pandas dataframe then wrap them that our feed should be correct. I attempted to
    use facet_grid but could not get the correct parameters working for the method. This
    resulted in incomplete code and not the desired result. I was only able to display a single
    hardcoded image with img_index before I ran out of time
    """
    print(gg)

if __name__ == '__main__':
    # run main
    main()
```

What my program was able to do at the end (not much different from the class demo at all)



```
❯ python3 src_v0.py
File already exists in this folder...continuing anyway

Shape of our converted array:  (2007, 256)

     col  row  intensity
0      0    0       -1.0
1      1    0       -1.0
2      2    0       -1.0
3      3    0       -1.0
4      4    0       -1.0
..   ...  ...        ...
251   11  -15       -1.0
252   12  -15       -1.0
253   13  -15       -1.0
254   14  -15       -1.0
255   15  -15       -1.0

[256 rows x 3 columns]
_
```

```python
"""
Homework 1 for Professor Hockings CS499-599 Deep Learning course
Author: Akiel Aries

Details:

"""
# lib for retrieving src file from web
import urllib.request
# lib for reading files on OS
import os
# lib used for copying src file info into destination
import shutil

import pandas as pd
import plotnine as p9
import numpy as np

# our src file we want to download
src_url = 'https://github.com/tdhock/cs570-spring-2022/raw/master/data/zip.test.gz'
src_file = 'zip.test.gz'

def main():
    """
    backbone code from the python.org lib docs to
        - check if a file exists in the current directory
        - retrieve a file given the url
    """
    if not os.path.isfile(src_file):
        urllib.request.urlretrieve(src_url, src_file)
        print("Downloading src file...")

    else:
        print("File already exists in this folder...continuing anyway\n")

    # read in downloaded src file as a pandas dataframe
    zip_df = pd.read_csv(src_file, header=None, sep=" ")

    """
    convert dataframe to numpy array exlcuding the first column; iloc for row
    and col specifying.
    """
    zip_arr = zip_df.iloc[:,1:].to_numpy()
    # printing shape of array
    print("Shape of our converted array: ", zip_arr.shape, "\n")

    """
    code for formatting important data values that correspond to imgs
    as well as looping thru first 9 rows and wrapping them into one grid
    """
    # define num of pixels
    pixels = 16
    index_vec = np.arange(pixels)
    # index of our images; arbitrary example to plot one of them
    img_index = 0
    zip_arr[img_index,:]

    img_df = pd.DataFrame({
        "col":np.tile(index_vec, pixels),
        "row":-np.repeat(index_vec, pixels),
        "intensity":zip_arr[img_index,:],
    })

    # print table of our values col, row, intensity
    print(img_df)

    """
    display plot using plotnine
    """
    gg = p9.ggplot()+p9.ggtitle("index")+p9.geom_raster(
        p9.aes(
src_v0.py                                                        1,1           Top
```

This program starts with some imports for our necessary libs and assigning the download url and file to variables respectively.

As we enter main there is an if else block saying if the file is already in the programs located directory, say that is already exists but continue anyways.

- I was unsure of how to read the entire file system without making messy code. Searching the entire file system for ‘zip.test.gz’ proves no worth since if we determine the file is already existing in the filesystem we are not tasked with reading in wherever the desired file is.
    - For example if ‘zip.test.gz’ is located in ~/Downloads but we are working in ~/Documents, detecting the src file would do no good unless we were to implement additional logic that would detect the file globally and use that src file in ~/Downloads from our development in ~/Documents instead of downloading that file again wherever this program is located. I am sure if I was able to take more time to figuring this logic out it could be done effectively!

After the conditional we read in the downloaded src file and store it in a pandas data frame. We don’t include the header and declare a separation of one space “ ”. To print the shape of the array we simply print and add in our array variable calling the shape attribute from there.

We then format the data values read in from the csv declaring a number of pixels, declaring a number for our index image and reading that from our numpy array.

From there we store the attributes for the images into a pandas dataframe to manipulate and display later. From here I printed out the image dataframe to display its some dimensions of our array.
We then go into using plotnine so we can finally display our plots. We make a gg variable and assign some different params ot it including the col row and intensity and finally assigning data to the img dataframe we made earlier. Finally we print the image.

- I had issues displaying more than one image in the plotline plot. I had tried to loop through the img indexes 0-9 and display a grid based off of that but was unsure what variables were to be passed around here. When I did display our images as a grid I got a very interesting looking plot of 16x16 displayed in a linear line with yellow, blue, purple colors all over the place. I decided to revert those changes and just display the given image as shown in the demo.

1. Print the shape attribute – how many rows/columns are there?

2007 rows, 256 columns

2. What is the number of rows/observations/example digits? What is the number of columns/features/pixels per example?

256 example digits and 16 features per example.