

Machine Learning Project 3 Neural Networks

Pedram Safaei, Ian Grant

December 2nd

1 compute_Z

1.1 Explanation

```
1 sample code
```

2 compute_covariance_matrix

2.1 Explanation

```
1 sample code
```

3 find_pcs

3.1 Explanation

```
1 sample code
```

4 project_data

4.1 Explanation

```
1 sample code
```

5 compress_images

5.1 Explanation

The `compress_images` function takes in a **DATA** array and an int **k** the number of principal components to use. It uses PCA to get the principal components to compress the images that were given as input by dotting the principal component with Z^* . Then using `reshape()` the image is returned to it's original aspect ratio instead of being flattened.

```
1 def compress_images(DATA,k):
2     exists = os.path.exists("Output")
3     if not exists:
4         os.mkdir('Output')
5     #for each pic in the data arr
6     Z = pca.compute_Z(DATA)
7     COV = pca.compute_covariance_matrix(Z)
8     L, PCS = pca.find_pcs(COV)
9     Zstar = pca.project_data(Z,PCS,L,k,0)
10    PCS = PCS[:, :k]
11    PCS = PCS.T
12    compress = np.dot(Zstar, PCS)
13    compress = compress.T
14    for j in range(0, len(compress)):
15        py.imshow('Output/out%d.png'%j,compress[j].reshape(60,48),vmin=0,vmax=255,cmap='gray',format='png')
```

6 load_data

6.1 Explanation

The load_data function takes in a **inputdir** it loads each image found in the directory into a temp var to then be flattened and loaded into the **data** var to be returned and used by compress_images.

```
1 def load_data(input_dir):
2     exists = os.path.exists(input_dir)
3     if not exists:
4         print("Can't find input dir")
5         return
6     dataimg = []
7     finalresult = []
8     input = input_dir
9     for dir, child, datas in os.walk(input):
10         for data in np.sort(datas):
11             image = py.imread(input+ data, 'pgm')
12             image = image.flatten()
13             dataimg.append(image)
14     finalresult = np.array(dataimg)
15     finalresult = finalresult.astype(np.float)
16     finalresult = finalresult.transpose()
17     return finalresult
```

7 Compressed images from DATA/TRAIN

7.1 10

7.2 100

7.3 500

7.4 1000

7.5 2000