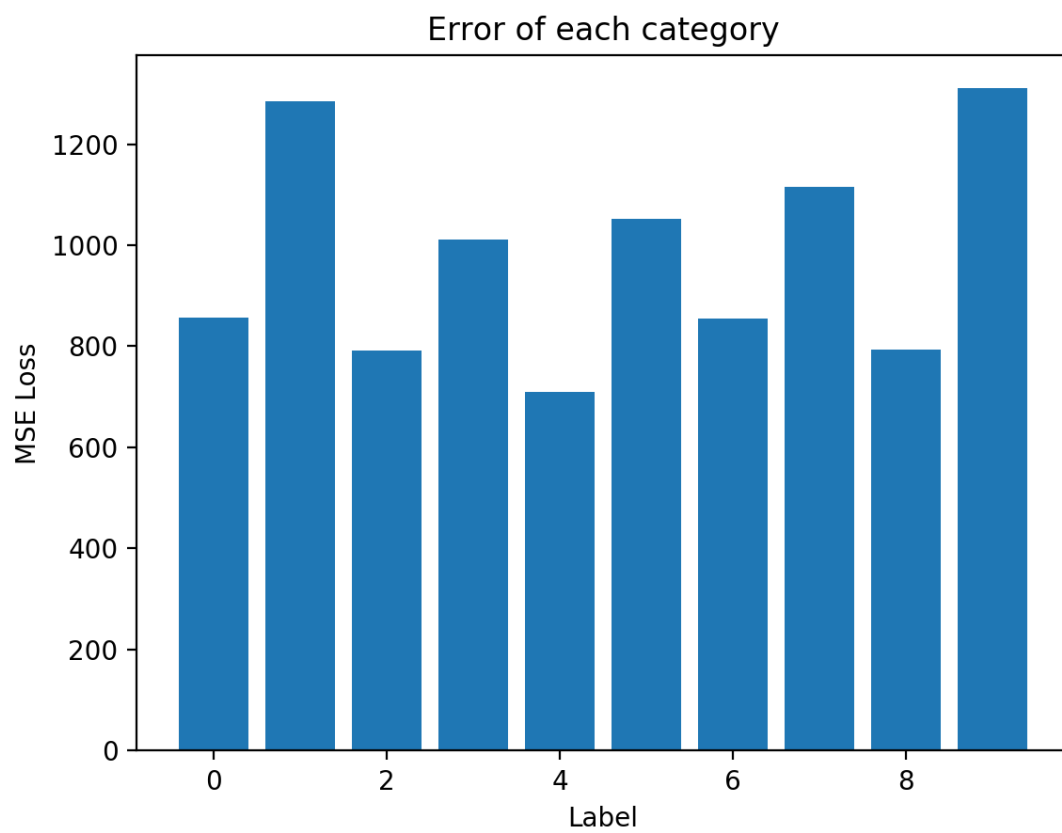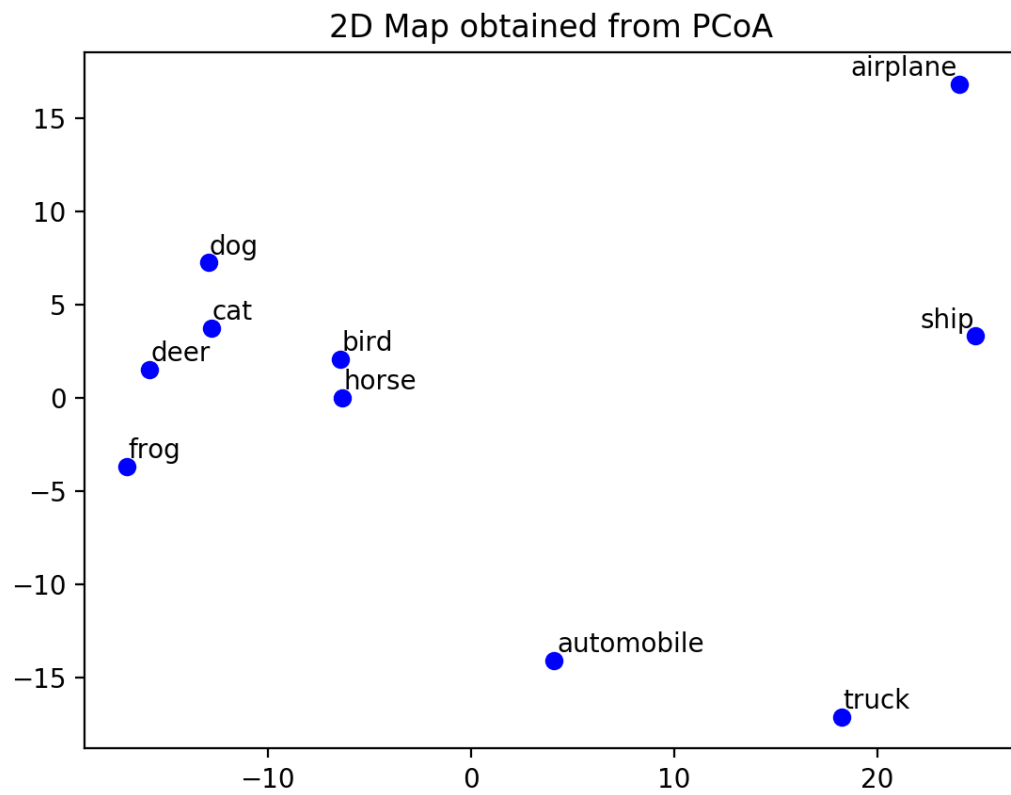# CS 498 AML HW4   Yu Che Wang/ yuchecw2

1. A single plot of error (use mean squared error) vs category (1-10)

2. A single 2D plot with the results of principal coordinate analysis applied to means

2D Map obtained from PCoA

Code:

```python
import numpy as np
from sklearn.decomposition import PCA
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
from scipy.spatial.distance import euclidean
# from skbio.stats.ordination import pcoa
from adjustText import adjust_text

def unpickle(file):
    import pickle
    with open(file, 'rb') as fo:
        dict = pickle.load(fo, encoding='bytes')
    return dict

meta_data = unpickle('cifar-10-batches-py/batches.meta')
data_file = ['cifar-10-batches-py/data_batch_%d'%i for i in range(1,6)]

# Key: label, value: a list of images with that label
label_data_dict = {i:[] for i in range(10)}
for file_num in range(1,6):
    original_dict = unpickle(data_file[file_num-1])
    for i, key in enumerate(original_dict[b'labels']):
        label_data_dict[key].append(original_dict[b'data'][i,:])

# Stack all images of a category into a single 2D numpy array
# Key: label, value: a matrix of images with that label
label_matrix_dict = {}
for i in range(10):
    label_matrix_dict[i] = np.empty((len(label_data_dict[i]), 3072))
    for j in range(len(label_data_dict[i])):
        label_matrix_dict[i][j,:] = label_data_dict[i][j]

mean_image_list = []
pca_list = [PCA(n_components=20) for i in range(10)]
recon_image_list = []
mse_list = []
for i in range(10):
    mean_image_list.append(np.mean(label_matrix_dict[i], axis=0))
    pca_list[i].fit(label_matrix_dict[i])
    recon_image_list.append(pca_list[i].inverse_transform(pca_list[i].transform(label_matrix_dict[i])))
    mse_list.append(mean_squared_error(label_matrix_dict[i], recon_image_list[i]))


plt.figure(1)
plt.bar(list(range(10)), mse_list)
plt.xlabel('Label')
plt.ylabel('MSE Loss')
plt.title('Error of each category')
plt.show()

distance_matrix = np.empty((10, 10))
for i in range(10):
    for j in range(10):
        distance_matrix[i,j] = euclidean(mean_image_list[i], mean_image_list[j])

# Principle Coordinate Analysis
A = np.identity(10) - 1/10 * np.ones((10,10))
W = -1/2 * np.dot(np.dot(A, distance_matrix), A.T)
eigen_value, eigen_vector = np.linalg.eig(W)
y = np.dot(eigen_vector[:,0:2], np.sqrt(np.diag(eigen_value[0:2])))

labels = np.array([str(x)[2:-1] for x in meta_data[b'label_names']])
fig, ax = plt.subplots()
plt.plot(y[:,0], y[:,1], 'bo')
texts = [plt.text(y[i,0], y[i,1], labels[i], ha='center', va='center') for i in range(labels.shape[0])]
adjust_text(texts)
plt.title('2D Map obtained from PCoA')
plt.show()
```