brunoc2
myronen2
abujaba2

Homework 3 - Applied Machine Learning

The dataset we will be dealing with in this homework is the CIFAR-10 dataset. This dataset is divided into five training batches and one test batch, each with 10000 images. In our codes we made sure to use all five training sets to come up with our results. The difficulty here is to deal with appropriate sizes and the corresponding RGB components and the way they are setup in the given dictionary that we are provided. It has been clearly noted and from what is seen in the code attached, that each class has it's own matrix of given size. We have modularized our code so as to be able to also get the image out of this by forming, through row-major order, an image of size 32x32x3. The particular function is "show_pic", many more functions have been defined for a much more pleasant reading of the code.

Part A

The code for this particular part of the question can be found in **"part-a.py"**.
We first start by computing the means of every class (for every batch of data). We came up with the following means:


Airplane

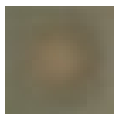

Automobile

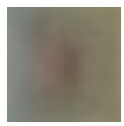

Bird


Cat


Deer


Dog


Frog


Horse


Ship


Truck

The latter clearly show distinct but rather blurry figures as well as colors, this is indeed what we are expecting as we are averaging a large amount of pictures one on top of the other (the image got blurrier as we added the batches). One example is that of the dog:

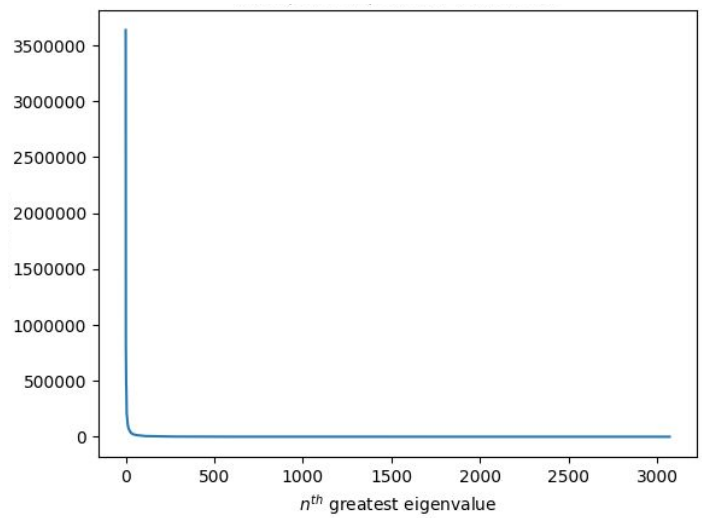

(mean dog with 1 batch)


(mean dog with 5 batches)

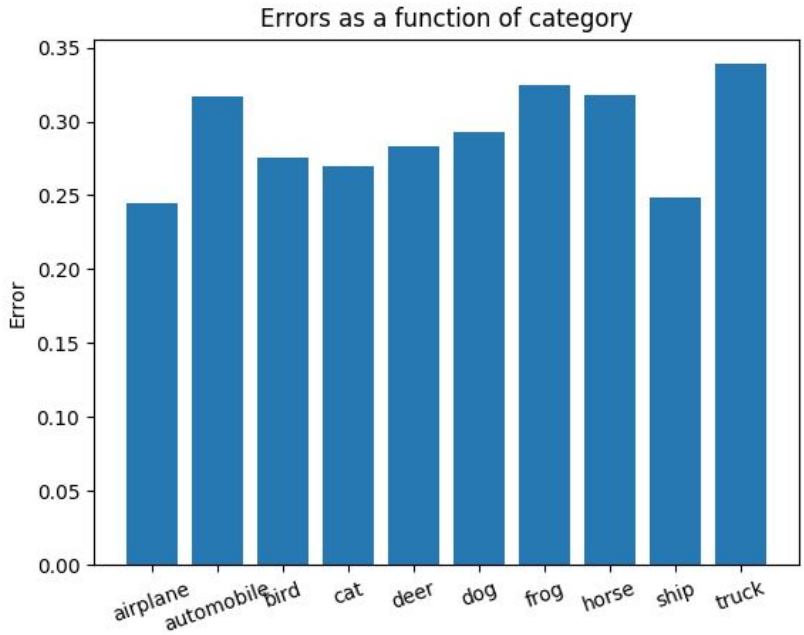
The next step is to compute the 20 first (with largest influence) principal components. This is easily done using our function "get_sorted_eigvec", the latter returns an ordered eigenvector matrix, that will allow us to rotate and expand our data correspondingly. We have also computed our eigenvalues to reach the aforementioned result. We have verified that they

brunoc2
myronen2
abujaba2

were indeed consistent with our expectations for all the different classes. We obtain a very similar curve for all the different classes (on all different batches) , as seen below:



It clearly shows that most of the information is indeed held by approximately the first 20-30 components. Hence the error when comparing an image formed by its first 20 principal components and the original image should be small. We compute this error as a next step in this process for the mean image of every class.



As can be seen from our bar chart, the error ranges from around 24% to 34% but of course this is dependant on what class we consider, our results thus seem to be as we predicted, a relatively low error for the selection of the first 20 principal components. This has been ran accumulating all batches from 1 to 5 as seen in the code.

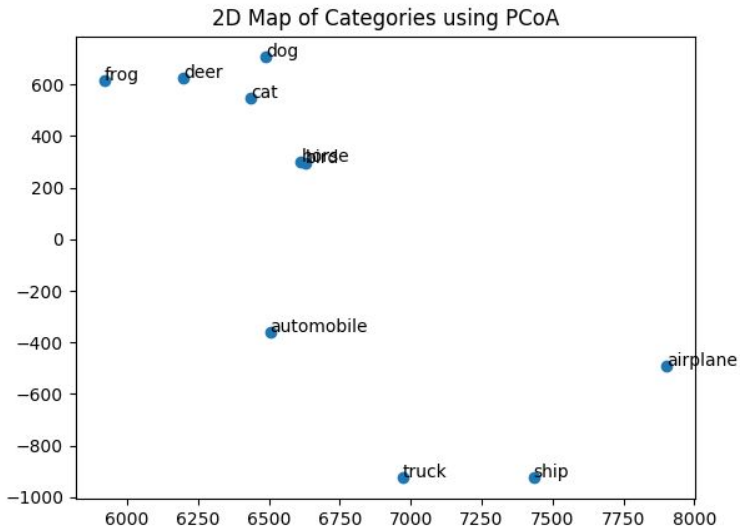
brunoc2
myronen2
abujaba2

Part B

The code for this particular part of the question can be found in **“part-b.py”**.
Here, we compute the distances between mean images for each pair of classes and then use principal coordinate analysis to make a 2D map of the means of each categories.
The different steps followed to compute the PCoA are closely linked to those that we have seen in class and in page 74 and 76 of the book. The main elements we initially had trouble with was the computation of matrix D, otherwise known here the distance between the mean images for each pair of classes.

The different steps followed to compute the PCoA have been extensively detailed in the comments of the code corresponding to this Part.

The resulting 2D map has been computed after using all of the batches:



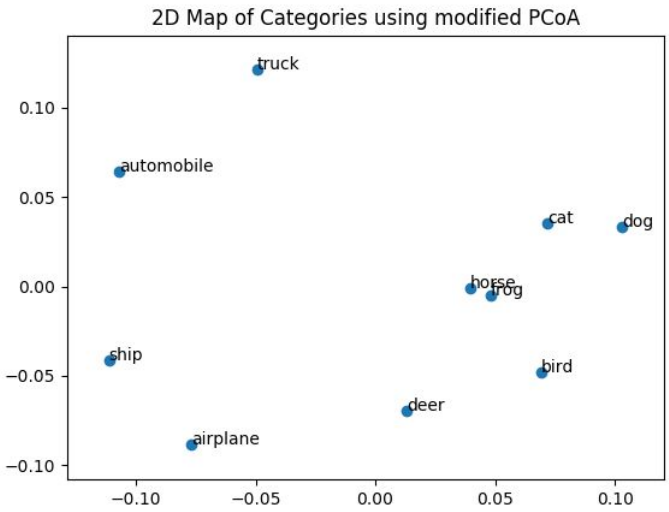
It is clear from this that we have obtained reasonably good results over all the batches. Indeed, it is clear from the map that we have somewhat of a clustering with respect to the means of the animals, hence the fact that they have a lesser distance on the map and thus more elements in common compared to machinery like automobiles , trucks, ships and airplanes. The next part will evaluate distances with respect to a different method involving the computation of the error and thus, intuitively, the similarity between the different classes in all the batches. Thus we expect this other method to provide us with a similar clustering but perhaps even better results in terms of distances between means as they are now a function of similarity with respect to error.

brunoc2
myronen2
abujaba2

Part C

The code for this particular part of the question can be found in **“part-c.py”**.
We now change our measures of similarity. For class A and class B we define E(A → B) to be the average error obtained by representing all the images of class A using the mean of class A and the first 20 principal components of class B. We now define the similarity between classes to be (1/2)(E(A → B)+E(B → A)).
We will use principal coordinate analysis once again to make a 2D map of the classes and compare the latter with the previous one seen in Part B.

Once again, the code clearly explains the different steps that have been taken, the latter are very similar to those seen in Part A and Part B, however obviously taking into account this new method of computing the 2D map.



The distance is now not being computed in the same way, explaining the fact that the different means are not positioned similarly as they previously were in Part B. However the this and the other results of this image are as expected: we still have a clustering with respect to the animals but this time the distances seem much more realistic (with respect to the given dataset). Moreover there is still the feature of the machinery being in their own individual spots rather than closely spaced means forming a cluster. Key observations include that the truck is now much farther away from the ship and rather closer to the automobile which is logical. The cat and the dog are also more closely related and the deer farther away from the rest. The only inconsistency that remains is that of the similarity between the horse and the frog most likely due to the color setting in which the photos have been taken. Other than this, both methods provide a good estimate of the distances between means and show the clustering we would and should expect.