Assignment 2 - K-Means Clustering and Autoencoders

Baasit Sharief

November 15, 2021

1 Dataset and Understanding the problem statement

1.1 Introduction

The goal of the first part of the assignment is to understand how a K-means clustering works from scratch to gain a better understanding of the intricacies of the algorithm.

The second part of the assignment is based on getting sparse embedding representation using Autoencoders and use the given embeddings to creates clusters.

For the whole assignment, we're dealing with CIFAR 10 dataset. The dataset consists of 50000 images of dimensions 32x32 for training and 10000 images of dimensions 32x32 for testing with 10 classes.

2 K-Means Clustering

2.1 Introduction

K-Means clustering is one of the most simplest and popular unsupervised learning algorithms. The objective of K-means is to group similar data points together and discover underlying patterns. To achieve it, we decide on a fixed number (k) of clusters in the dataset, which determines the number of centroids in the dataset. Every data point is then allocated to each of the clusters through reducing the in-cluster sum of squares. K-means identifies the k number of centroids and then allocates every data point to the nearest cluster based on equation (1). We iteratively recalculate the centroids until we reach a convergence based on equation (2).

$$C_i^{(t)} = \left\{ x_p : \left\| x_p - a_i^{(t)} \right\|^2 \leq \left\| x_p - a_j^{(t)} \right\|^2 \ \forall j, 1 \leq j \leq k \right\} \tag{1}$$

$$a_i^{(t+1)} = \frac{1}{|C_i^{(t)}|} \sum_{x_j \in C_i^{(t)}} x_j \tag{2}$$

Algorithm 1 K-Means Clustering

Require: $k \geq 1, x, N$
Ensure: $x.shape = 2$
  $a \leftarrow a_1, a_2, a_3, ..., a_k$ ▷ Initialize centroids
  $dist \leftarrow distance(x, a_i)$ ▷ Calculate distance
  $labels \leftarrow C(x)$ ▷ Store labels
  while $N \neq 0$ **do**
    for $i \leftarrow 1$ to k **do**
      $a_i \leftarrow \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j$ ▷ Update centroids
    end for
    $dist \leftarrow distance(x, a_i)$ ▷ Recalculate distance
    $labels \leftarrow C(x)$ ▷ Store labels
    $N \leftarrow N - 1$
  end while

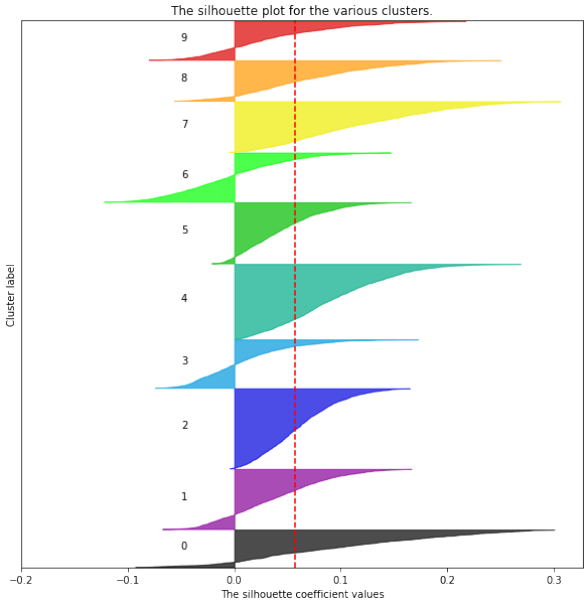
2.2 Experimental Setup and Results

The dataset has 10 classes so we run K-means clustering on the value of k = 10. With the help of python libraries like NumPy, and Scipy along with JupyTer Notebook, we have implemented K-means clustering. Before doing the clustering, We scale the points to [0,1] by dividing each point with 255. We run the algorithm for different number of iterations i.e. 100, 300, 500, 1000 and reach convergence in 100 iterations itself.

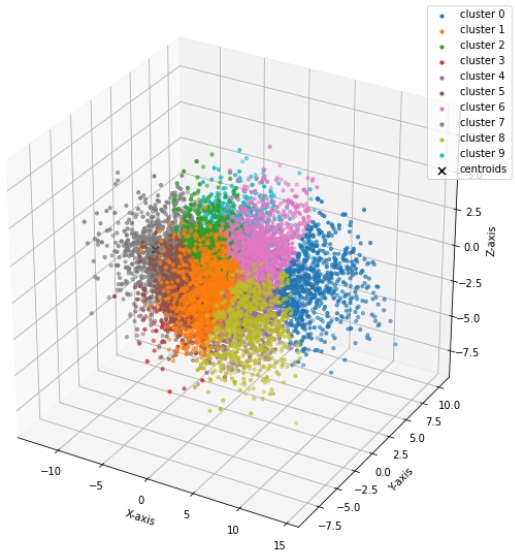
Below are the observed results on the given dataset.

Iterations	ASC	Dunn Index
100	0.0568	0.0927
300	0.0568	0.0927
500	0.0568	0.0927
1000	0.0568	0.0927

Given below is the silhouette plot and the 3-D scatter plot of the data points. The data points were decomposed into 3 features using PCA.



(a) Silhouette Plot



(b) 3-D Scatter Plot

Figure 1: Cluster Analysis

3 Autoencoders

3.1 Introduction

Autoencoders are a type of neural networks which used to learn sparse representations of instances from any unstructured data. These encodings are usually validated by regenerating the input from these encodings.

Autoencoders architecture usually involves an encoder and decoder which are trained together. The encoder is used to create sparse representations of the data and decoder is used for regeneration of input and validation of the encoder output.

3.2 Experimental Setup and Results

For the autoencoder, I had experimented with different architectures and went with Convolutional Autoencoders because we're dealing with an image dataset.

After trying several architectures for the encoder I decided to go forward with 3 convolutional layers followed by maxpooling layers to reduce the dimensions. For the decoder, I went with a similar architecture but instead of maxpooling i went with upsampling to get the same dimensions as the output. Below is training loss plot for the autoencoder and reconstructed images.

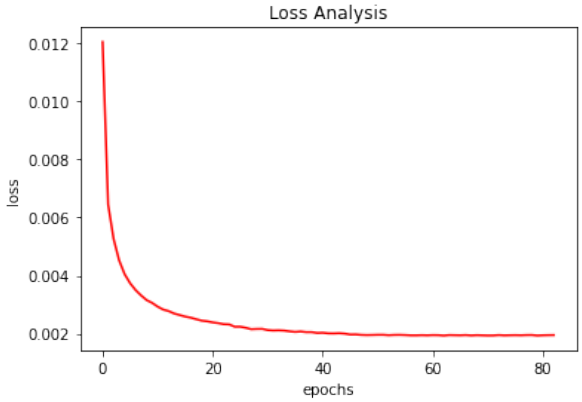
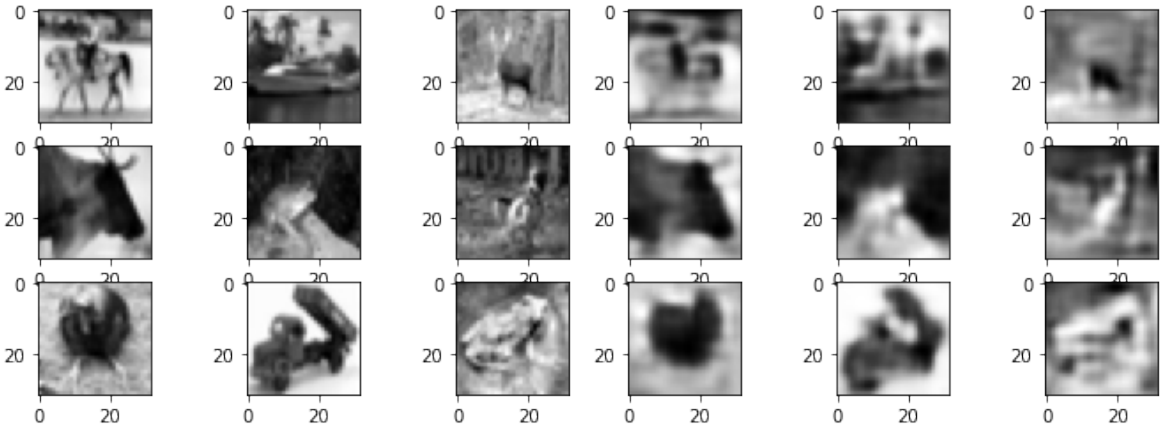


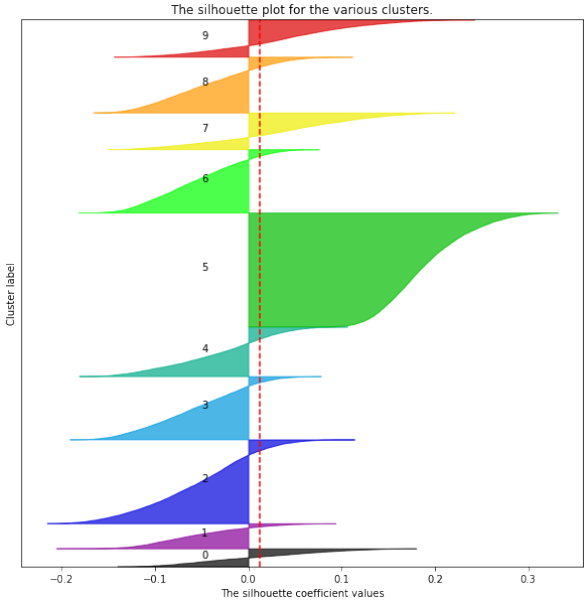
Figure 2: Autoencoder loss



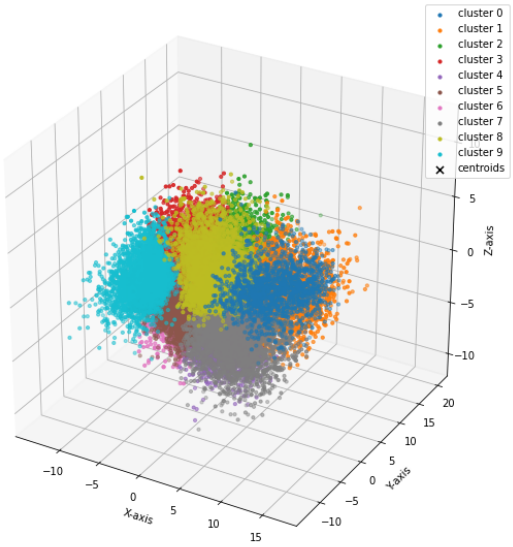
(a) Original Input (b) Reconstructed Input

Figure 3: Autoencoder: Reconstruction

Following is the visualizations of the results achieved by the autoencoder.



(a) Autoencoder: Silhouette Plot



(b) Autoencoder: 3-D Scatter Plot

Figure 4: Autoencoder: Cluster Analysis

We got an ASC score of 0.01253.

4 Conclusion

Through the assignment we were introduced to clustering methods like K-means as well as the basic idea of of how Autoencoders are built and used for sparse representation use cases.