Assignment 2 - CSE 574 - Introduction to Machine Learning

Adeline Grace George

November 14, 2021

1 Assignment Overview

The aim of the assignment is to perform unsupervised learning on the Cifar 10 dataset. This is done by implementing K-means clustering on the raw data from scratch.

2 Dataset

The Cifar 10 dataset has a total of 60,000 examples split into training set of 50,000 examples and a test set of 10,000 examples. Each example is a 32x32 image, associated with a label from 10 classes. Each image is 32 pixels in height and 32 pixels in width, for a total of 1024 pixels in total. This pixel-value is an integer between 0 and 255. The training and test data sets have 1025 columns including the labels.

3 Python

Jupyter Notebook has been used for implementation.

3.1 K-Means Clustering Implementation - Part 1

In the given dataset (Cifar 10), each of the example images are associated a label from 10 classes (airplane, autombile, bird, cat, deer, dog, frog, horse, ship and truck). The goal is to find homogeneous sub-groups and split the data into clusters.

3.1.1 Parameters and Equations used

This is done in the following steps:

1. After image preprocessings, wherein they are converted to grayscale images and normalized, an arbitrary number of clusters are chosen and centroids

are picked randomly (one for each cluster).

$\mathbf{K} \rightarrow$ Number of clusters

$\boldsymbol{\mu_1, \mu_2, ..., \mu_k} \in \mathbf{R} \rightarrow$ Randomly chosen centroids

2. For every residual datapoint, the euclidean-based distance of the datapoint to the chosen centroids is calculated and the datapoint is assigned to the cluster with the minimum euclidean distance, thus making sure that datapoints in a cluster are as similar as possible.

Let $\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_m} \rightarrow$ be the training set

Euclidean distance is given by: $\|\mathbf{x_i} - \boldsymbol{\mu_i}\|$

3. The mean value of each cluster is chosen as the new set of centroids.

For k = 1 to K,
$\boldsymbol{\mu_k} \rightarrow$ average (mean) of points assigned to cluster 'k'

4. This process is repeated for a chosen number of iterations until optimum clustering is achieved.

3.1.2 Visualization

The below graph shows how the cost decreases with increasing iterations:

3.1.3 Results

Average Silhouette Coefficient (ASC): 0.055633247
Dunn Index: 0.090204

3.2 Auto-Encoder Implementation - Part 2

An Autoencoder is an optimization process that follows two steps: Compression of input data into a lower dimension and reconstruction of the lower dimensional data to recreate the original input. The aim is to preserve essential features by eliminating unecessary details. The difference between the original input and the reconstructed data is termed as reconstruction error and the aim is to reduce this error to a minumum while training the autoencoder model. The model exploits the natural structure in the data to find an efficient way to represent it in a meaningful lower dimensional space.

3.2.1 Encoding

Encoding is the process of flattening the original data into a lower dimensional representation. It is different from a simple grascale flattening in the sense that

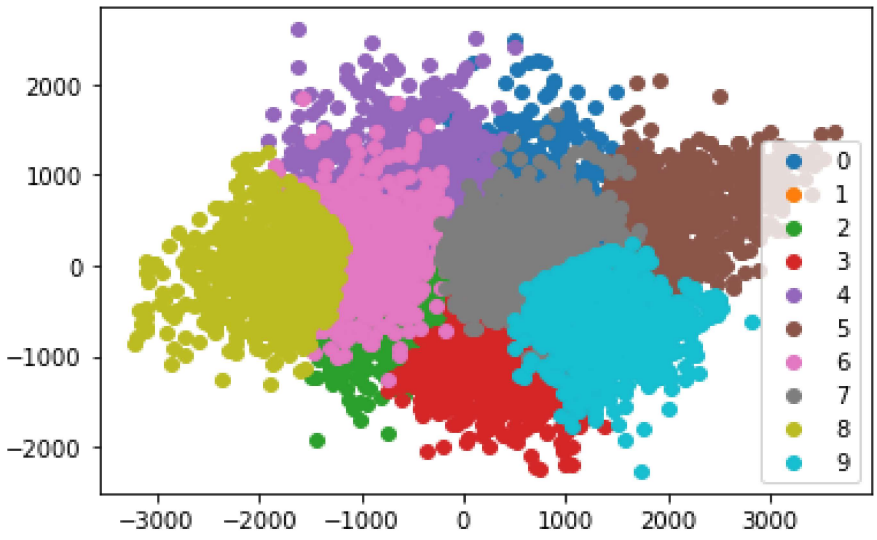

Figure 1: K-Means Cluster Plot

it preserves a "meaningful" description of the data. The train dataset has images with 32 x 32 in pixel values, so the data contains 1024 values. The aim of the encoder is to compress this data so the classifier will only handle necessary data.

The encoder begins with an input layer with shape (32 x 32 x 1). This is immediately flattened to 1024 values which is used to created a fully connected dense layer consisting of merely 64 values (approximately 6% of the original data) using the **relu** activation function. This is the end of the encoder model.

3.2.2 Decoding

Decoding comes after encoding and they together comprise the auto-encoder. Decoder is considered to be a mirror implementation of the encoder. So a dense layer of 64 values is created using the same activation function that was used while encoding - **relu**. To retrieve the original image, another dense layer is created with 1024 values. The final decoder output is a reshaped array of dimensions (32 x 32 x 1). This marks the end of the decoder model.

This is followed by an implementation of the **Adam** optimizer with learning rate of 0.01 and a decay of 0.000001.

Both the encoder model and the decoder model are combined to form the autoencoder model and the training summary is displayed. The autoencoder

model is compiled with the optimizer and mean squared error is taken to be the loss metric. The number of epochs is set to 3 and the model is saved for each iteration.

3.2.3 Results

Average Silhouette Coefficient (ASC): 0.079234
Dunn Index: 0.045633