Unsupervised Learning on Cifar 10 dataset

Dexson John D'Souza
Department of Computer Science
University at Buffalo
Buffalo, NY 14260
dexsonjo@buffalo.edu

1. Project Overview

For part 1, we will perform K-means clustering on Cifar 10 dataset. We will also perform qualify analysis of our result using Average Silhouette Coefficient and Dunn's Index evaluation metrics.

For part 2, we will use Auto-Encoder on the dataset. Using Auto-encoder, we will generate a sparse representation of the dataset. We perform K-means clustering on the encoded data. We will use Average Silhouette Coefficient to assess the quality of the clusters.

2. Experimental Setup

We have used Jupyter Notebook which is an open-source software. We have used CV2 library to perform gray scale conversion. We have used Scikit-learn which is open-source software for machine learning library in Python programming language to perform data partitioning and data normalization. We have used Numpy which is library for the Python programming language. We have also used the Python library Pandas. We have use Matplotlib to plot the clusters. We have used Keras to perform Auto-Encoding. We have used sklearns to perform k-means clustering on the encoded data in part 2.

3. Result Analysis

3.1 K-Means Clustering(Part 1)

3.1.1 Overview and Result

In this phase, we implemented K-means Clustering from scratch. 10,000 testing sample of Cifar10 dataset was used. We have used cv2 library to convert the colored images in our dataset into gray scale images. We have performed normalization on the data. We have used Principal Component Analysis to reduce the dimension of data from (10000,1024) to (10000, 2). We have performed k-means clusters with 10 clusters and 50 iterations. The clusters formed are depicted in figure 1.

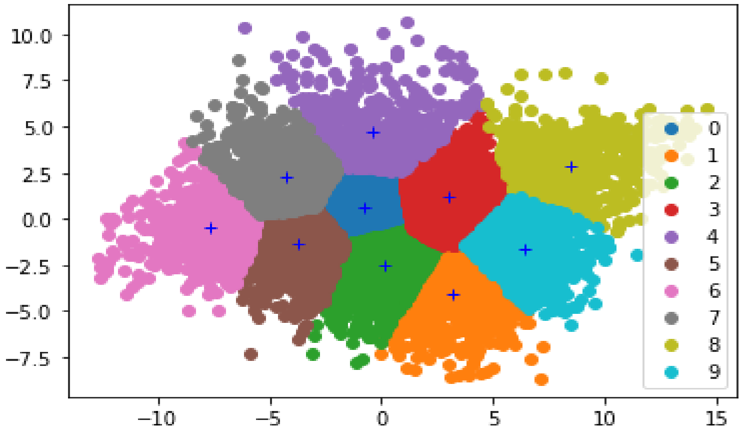

Fig 1: Clusters formed by Kmeans Clustering Algorithm

3.1.2 Silhouette Coefficient Calculation

Silhouette analysis is used to study the separation distance between the final clusters. The silhouette value is a measure of how similar an image is to its own cluster compared to other clusters. We have used Sklearn to perform Silhouette analysis on our result. The score obtained using Silhouette analysis was 0.32305557.

3.1.3 Dunn's Index Calculation

Dunn's Index is a metric used for cluster analysis. The goal is to find the difference between members of difference clusters. We have used Validclust library to calculate Dunn's Index. The value obtained was 0.1518.

3.1.4 Dunn's Index Analysis

Quality analysis of the K-means algorithm for different no. of clusters is depicted in Table 1.

Table 1: Dunn's Index for different no. of clusters.

No of Clusters	Dunn's Index
6	0.1377
7	0.1387
8	0.1409
9	0.1457
10	0.1518

Graph plot of Dunn's Index for different no of clusters is depicted in Fig 2

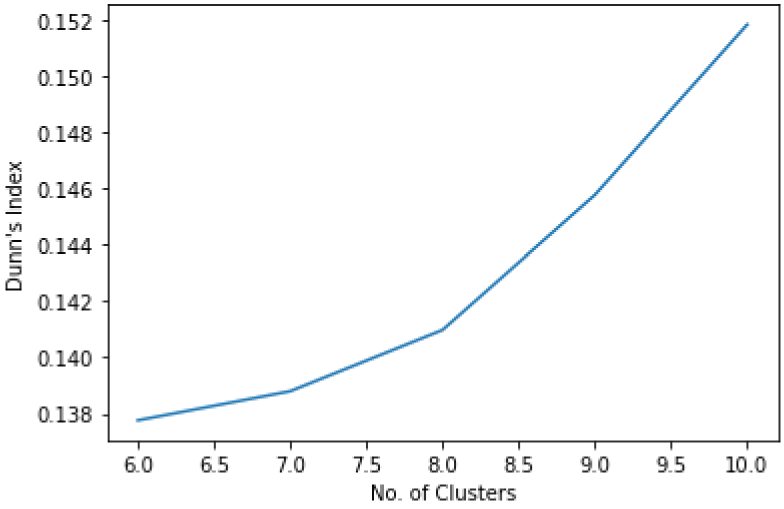

Fig 2: Dunn’s Index vs No. of Clusters

3.2 Auto Encoder Implementation(Part 2)

3.2.1 Overview

In this phase, we have used 50,000 training samples of Cifar-10 dataset. We have performed normalization on the dataset. We have added noise to our input to improve our encoder’s performance. We have used Keras to create the encoder and decoder.. We have used sklearn’s Kmeans to perform k-means clustering with 10 clusters and random_state 1234. We have performed quality analysis on the clusters.

3.2.2 Auto-Encoding for Cifar-10 dataset

We have created a Neural Network with 10 layers to perform Auto-Encoding. Our encoder consists of four 2D-Convolution layers and 2 BatchNormaliztion layers. Our decoder consists of one UpSampling layer, one Batch Normalization layer and two 2D-Convolution layers.

3.2.3 Auto-Encoding for Cifar-10 dataset

We have trained the model for 10 epochs and batch size of 500. We have used Adam Optimizer for learning purposes. We have used Binary Crossentropy for loss calculations. We were able to obtain a loss of 0.5530.

3.2.4 Silhouette Coefficient Calculation

The score obtained using Silhouette analysis was 0.51838267.

3.2.5 Result of AuoEncoding

We have used the custom noisy input data to train our model. Our model was able to denoise the noisy input successfully, this is depicted in figure

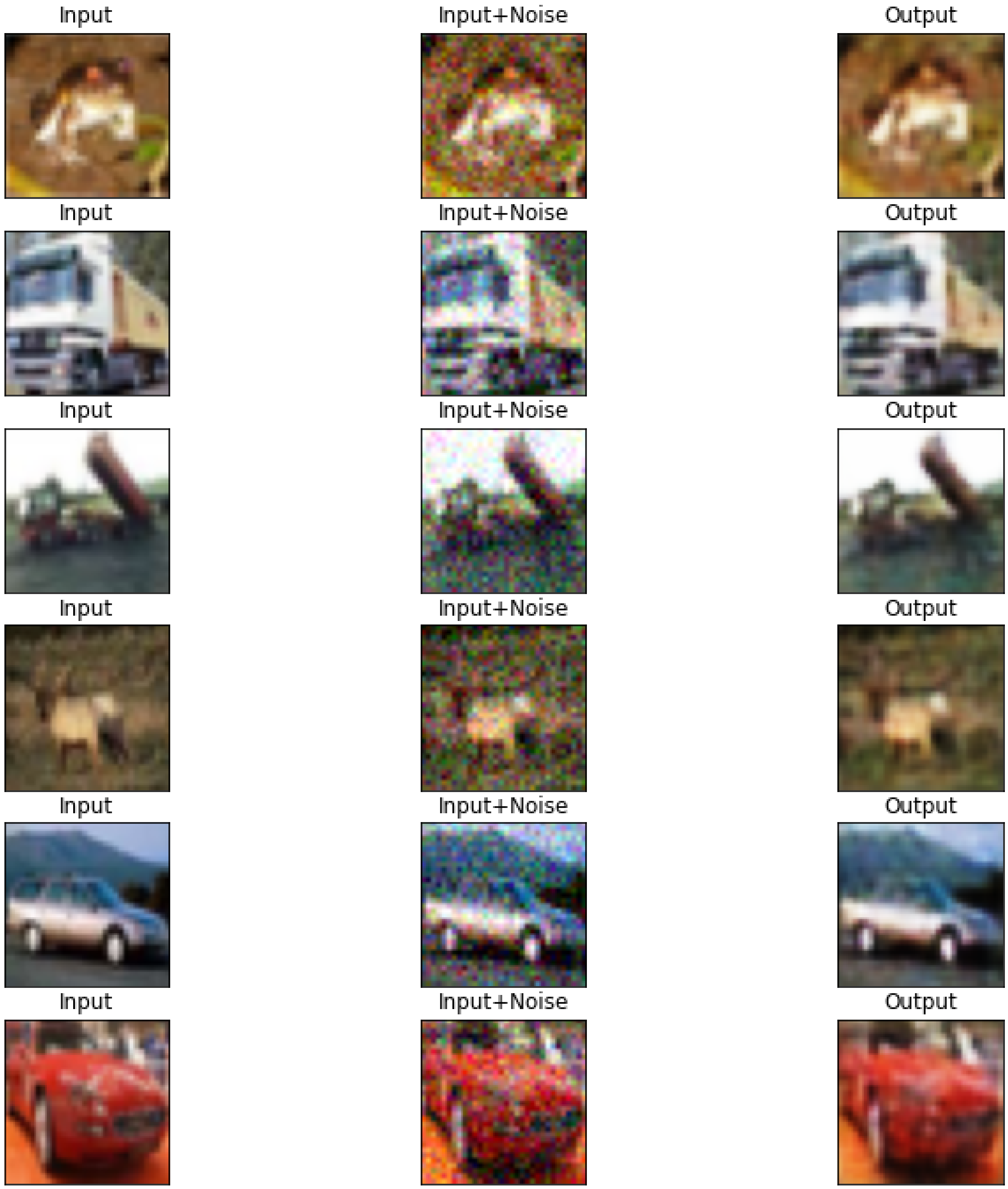

Fig 3: Comparison of AutoEncoder Output.

Our model was able to generate images which are similar to the original dataset using the noisy input. The encoded data used for Kmeans was able to generate a fine Silhouette Score. Hence, we can see how Autocoders can be used for Dimensionality Reduction as well as for Denoising Input Images.