Assignment 2
Clustering

RanjithKumar Hiremath

1. Objective:

This project has two main objectives Objective of this project (for this checkpoint) is to implement k-means clustering algorithm from scratch and evaluate the quality of clusters using **average silhouette score** and **Dunn's Index.** In part - 2 we try to build a auto encoder where in we extract the encoded representation of the image to cluster the image data.

2. Data Set:

Data set used for this project is **CIFAR-10,** which has a train data set of 50000 images and test data of 10000 images with 10 target classes. The dimensions of the images is 32 * 32, with all channels(RGB). For the purpose of k-means clustering, we only use the test data set with 10000 images.

3. Pre-Processing:

Since we are doing clustering, we perform the following pre-processing steps.

1. Convert the coloured image to gray scale.
2. Flatten the numpy arrays to shape 1024.
3. Normalise the arrays by dividing every intensity value by 255.

4. Clustering:

Clustering is done using the following steps.

I. Initialise the number of clusters k.
II. Initialise cluster centres randomly.
III. Calculate distance of each point to every cluster centres.
IV. Assign the point to the shortest distance cluster.
V. Re calculate the cluster centres.
VI. Repeat from step III.
VII. Stop after there is no change in the clusters.

Since we know that test data set has 10 target classes. Initialising the k to 10 and performing the clustering operation.

Evaluation of cluster quality is done using 2 metrics.

1. Average silhouette score:

The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. The Silhouette Coefficient for a sample is (b - a) / max(a, b).

Average of all these scores is Average Silhouette score.

For 10 clusters the silhouette score is **0.054**.

2. Dunn's Index:

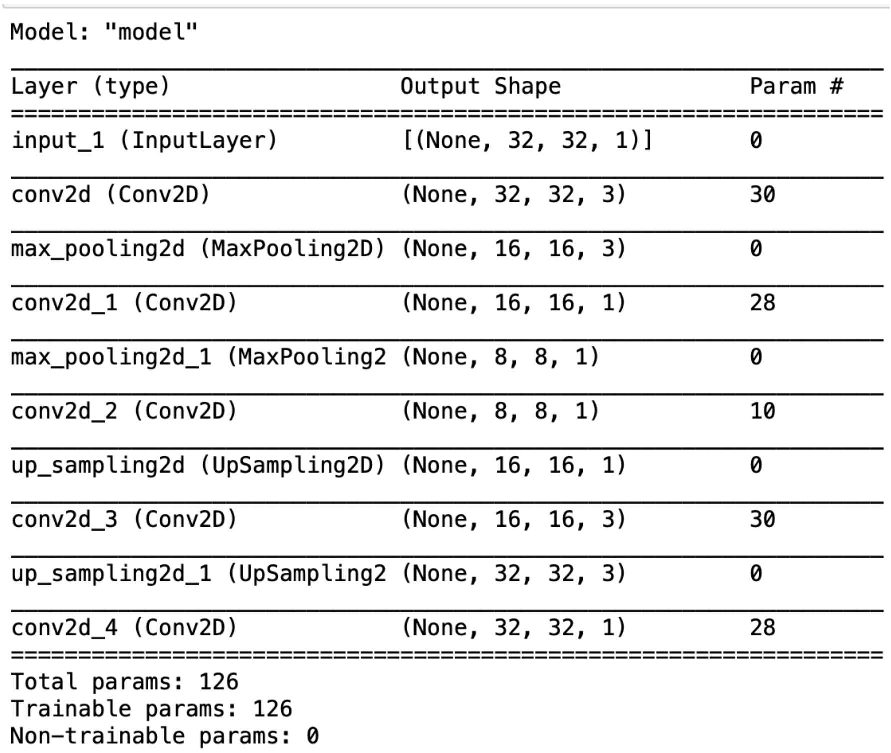
Dunn's index is the ration between the smallest distance between and observations not in the same cluster to the largest intra cluster distance.

$$DI_m = \frac{\min\limits_{1\leqslant i<j\leqslant m} \delta(C_i, C_j)}{\max\limits_{1\leqslant k\leqslant m} \Delta_k}$$

For 10 clusters the Dunn's index value is 0.0899.

5. **Auto encoders**:

Auto encoders are a class of models where we encode the input data into different features, such that these encoded features can be used to reconstruct the original input with minimum errors.



```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_1 (InputLayer)         [(None, 32, 32, 1)]       0         
_________________________________________________________________
conv2d (Conv2D)              (None, 32, 32, 3)         30        
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 16, 16, 3)         0         
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 16, 16, 1)         28        
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 8, 8, 1)           0         
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 8, 8, 1)           10        
_________________________________________________________________
up_sampling2d (UpSampling2D) (None, 16, 16, 1)         0         
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 3)         30        
_________________________________________________________________
up_sampling2d_1 (UpSampling2 (None, 32, 32, 3)         0         
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 32, 32, 1)         28        
=================================================================
Total params: 126
Trainable params: 126
Non-trainable params: 0
```

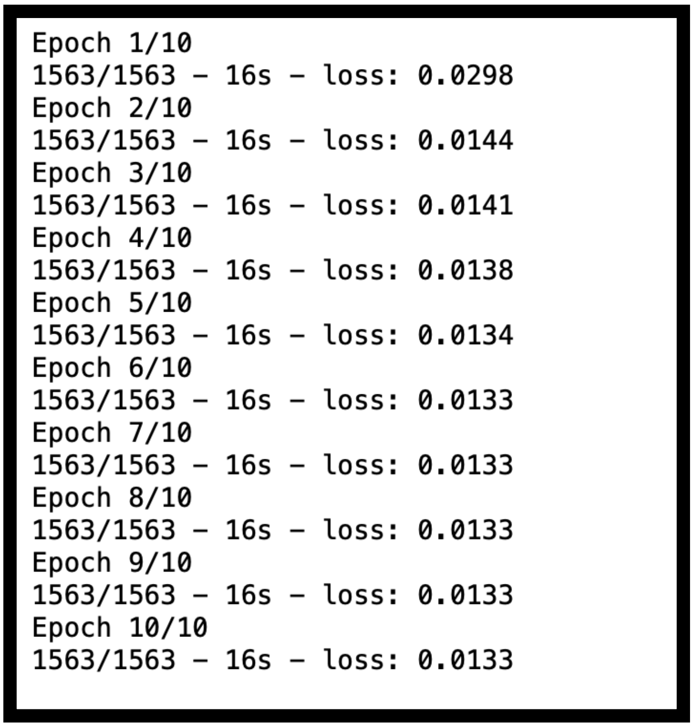
For this task the model architecture used is depicted in the above screenshot.

The layers does the following tasks.

1. Input layer accept the input as it is which is 32 * 32 image.
2. First Convolution layer has 3 filters, with padding “same” which keeps the output image shape same, irrespective of the size of the filter.
3. Max pooling layer with filter size 2, picks the max pixel value in 2 * 2 area of the convoluted image. This reduces the size of the image by half.
4. The next convolution and max pooling layer perform the same function. The final output of max pooling layer is an 8 * 8 image. This is where the encoder part ends.
5. The next layers signify the operations done on the image in reverse order, Where instead of max pooling upsampling is used and in the final layer, the input image is re constructed.

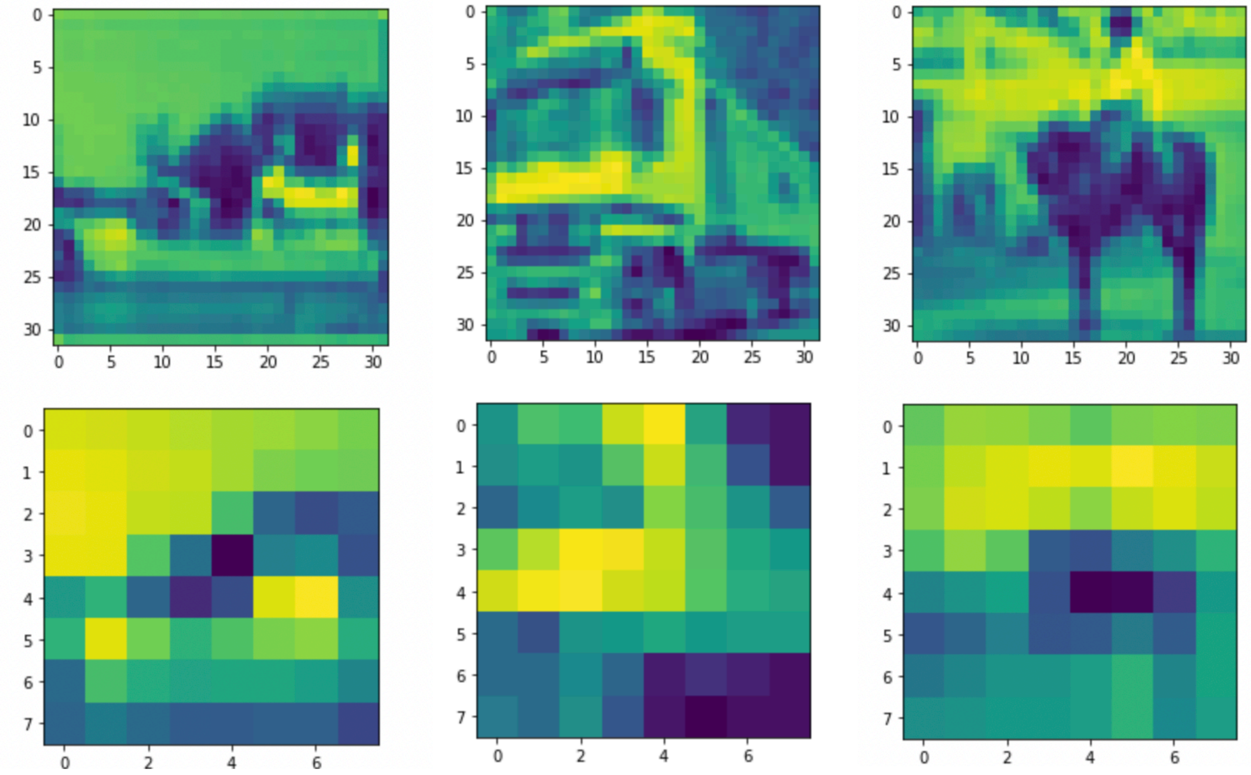
The Original image and the image reconstructed are compared using mean squared error as loss function. Our goal of training is to reduce mean squared error to get good reconstructed images, which indirectly signify good encoded representation.

Training the model on train of CIFAR for 10 epochs with a batch size of 32, results in the following loss values. Loss values are quite stable by epoch 8 and there is no further decrease in loss.



```
Epoch 1/10
1563/1563 - 16s - loss: 0.0298
Epoch 2/10
1563/1563 - 16s - loss: 0.0144
Epoch 3/10
1563/1563 - 16s - loss: 0.0141
Epoch 4/10
1563/1563 - 16s - loss: 0.0138
Epoch 5/10
1563/1563 - 16s - loss: 0.0134
Epoch 6/10
1563/1563 - 16s - loss: 0.0133
Epoch 7/10
1563/1563 - 16s - loss: 0.0133
Epoch 8/10
1563/1563 - 16s - loss: 0.0133
Epoch 9/10
1563/1563 - 16s - loss: 0.0133
Epoch 10/10
1563/1563 - 16s - loss: 0.0133
```

In the next step using only the encoder part of the model to get representations of the training set. Few examples of the original and encoded images are as shown below.



Using the encoded representation on the clustering algorithm described in 4. The ASC score is found to be **0.104**0 and Dunn's index is **0.02**. with 10 clusters.