

ASSIGNMENT: 3

Assignment Document

edureka!

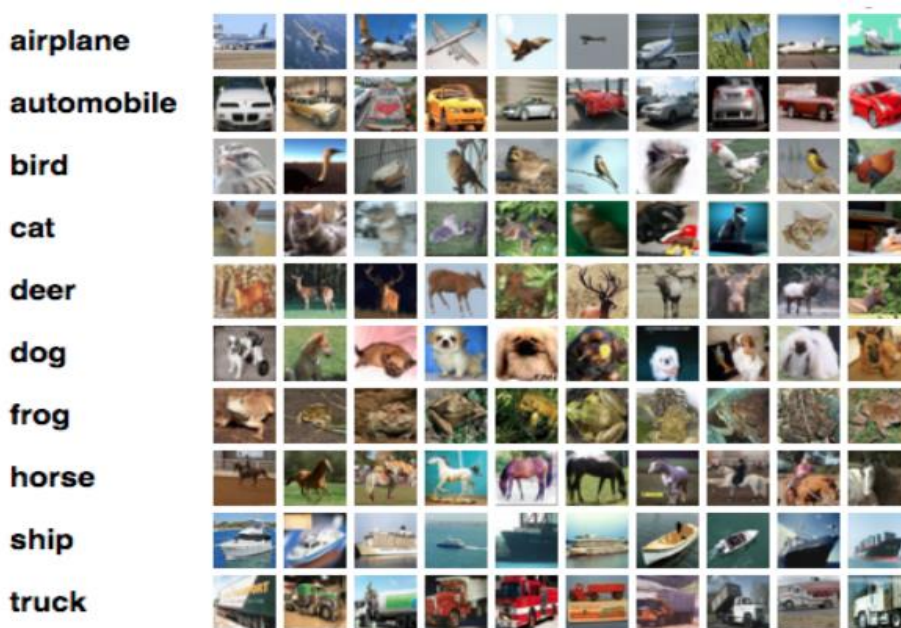
edureka!

© Brain4ce Education Solutions Pvt. Ltd.

Scenario – 1

Problem Statement:

The CIFAR-10 dataset, the goal of this problem is to run autoencoder on CIFAR10 dataset and see the reconstructed images using AE. Solve all the questions.



Dataset:

Total Images :- 60,000

Train Images :- 50,000

Test Images :- 10,000

Images/ Class :- 6,000

Image Size :- 32 X 32

Different Classes:

- Classes: 'Airplane', 'Automobiler', 'Bird', 'Cat', 'Deer', 'Dog', 'Frog', 'Horse', 'Ship', 'Truck '

Note: Please use google colab to work on this project. Also, make sure to select GPU backend while selecting a runtime.

<https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d>

Question 1:

A. Load CIFAR-10 data from Keras Library. And Split the same into Train and Test.

Output :-

Downloading data from <https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz>
170500096/170498071 [=====] - 2s 0us/step

B. Normalize the Data

HINT:-

- take the X train and convert it into float32
- Normalize the data with dividing pixel values with 255.

QUESTION 2:

A. Define the Encoder

HINT:-

- Input shape be: (32,32,3)
- First ConvLayer:
 - Conv2D Layer with 64 Filters of (3,3)
 - BatchNormalization layer

- Relu Activation
- 2D MaxpoolingLayer with (2,2) filter
- Second Layer
 - Conv2d layer with 16 filters (3,3)
 - BatchNormalization layer
 - Relu Activation
- Final Encoded as MaxPool with (2,2) with all previous layers

B. Define the Decoder as:

HINT:-

- Input shape be: encoder output
- First ConvLayer:
 - Conv2D Layer with 16 Filters of (3,3)
 - BatchNormalization layer
 - Relu Activation
 - UpSampling2D with (2,2) filter
- Second ConvLayer
 - Conv2D Layer with 32 Filters of (3,3)
 - BatchNormalization layer
 - Relu Activation
 - UpSampling2D with (2,2) filter

- Final Decoded as Sigmoid with all previous layers

Question 3:

Compile and fit model:

HINT:-

- Model:
- Pass input_img, decoded
 - model = Model(input_img, decoded)
- Compile:
- Optimizer: "adam"
- - loss = "binary_crossentropy"
- - Fit:
- - batch_size = 1000
- - epochs = 50

Train on 50000 samples, validate on 7000 samples

Epoch 1/3

50000/50000 [=====] - 42s 843us/step - loss: 0.5829 - val_loss: 0.5748

Epoch 2/3

50000/50000 [=====] - 36s 711us/step - loss: 0.5726 - val_loss: 0.5699

Epoch 3/3

50000/50000 [=====] - 35s 706us/step - loss: 0.5698 - val_loss: 0.5691