<https://colab.research.google.com/drive/1Saanzd_9UmAlz8PukKKLTJi1bq83VcvC?usp=sharing>

CIFAR-10

<https://colab.research.google.com/drive/1ZBQs6Xj1KxgbqPsMtbeCmt4UTVcDJfO0#scrollTo=cuJsAwGbmt_4>

**CIFAR-10 Dataset Description of codes using tensorflow.keras- CNN model \_\_\_\_**

Step 1-  import necessary libraries from tensorflow.keras like layers, models

Step 2- import CIFAR-10 dataset

Step 3- Normalize pixel values to be between 0 and 1 (divided by 255 for images)

Step 4- train dataset size is 50000, shape is 32\*32 and 3 for color image

Step 5 - we convert train image as float values

Step 6 - test dataset size is 10000, shape is 32\*32 and 3 for color image

Step 7 - we convert test image as float values

Step 8 - we convert output as categorical

Step 9 - we add Convelutional2D layer and we use 32 as filter size and 3 for color images

model.add(layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(32, 32, 3)))   model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Conv2D(64, (3, 3), activation='relu'))        model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Conv2D(64, (3, 3), activation='relu'))

Here 32 is filter size in input layer and then 64 is input fot hidden layer in Convolutional2D layer….

Step 10 - we use Relu Activation function for output layer

Step 11 - it converts 2D to 1D

Step 12 - we compile model as optimizer adam and loss= categorical\_crossentropy

Step 13- we train a model on train dataset

Step 14 - Evaluate the test accuracy

Step 15 - test accuracy is 100%

Step 16 - Evaluate the train accuracy

Step 17 - train accuracy is 100%