Artificial neural Network

import tensorflow as tf

from tensorflow.keras import datasets,layers,models

import matplotlib.pyplot as plt

import numpy as np

(x\_train,y\_train),(x\_test,y\_test) = datasets.cifar10.load\_data()

x\_train.shape

y\_train = y\_train.reshape(-1,)

y\_train[ :5]

x\_train = x\_train/255

x\_test = x\_test/255

ann = models.Sequential([

layers.Flatten(input\_shape=(32,32,3)),

layers.Dense(3000, activation="relu"),

layers.Dense(1000, activation="relu"),

layers.Dense(10, activation="sigmoid")

])

ann.compile(optimizer='SGD',

loss='sparse\_categorical\_crossentropy',

metrics=['accuracy'])

ann.fit(x\_train, y\_train, epochs=7)

ann.evaluate(x\_test, y\_test)

classes = ['aeroplane','automobile','bird','cat','deer','dog','frog','horse','ship','truck']

def true\_samp(x,y,index):

plt.figure(figsize =(15,2))

plt.imshow(x[index])

plt.xlabel(classes[y[index]])

y\_test = y\_test.reshape(-1,)

y\_pred = ann.predict(x\_test)

y\_pred[ :5]

y\_classes = [np.argmax(element) for element in y\_pred]

y\_classes[:5]

y\_test[ :5]

#Actual image

true\_samp(x\_test,y\_test,1)

#pridicted image

classes[y\_classes[1]]