

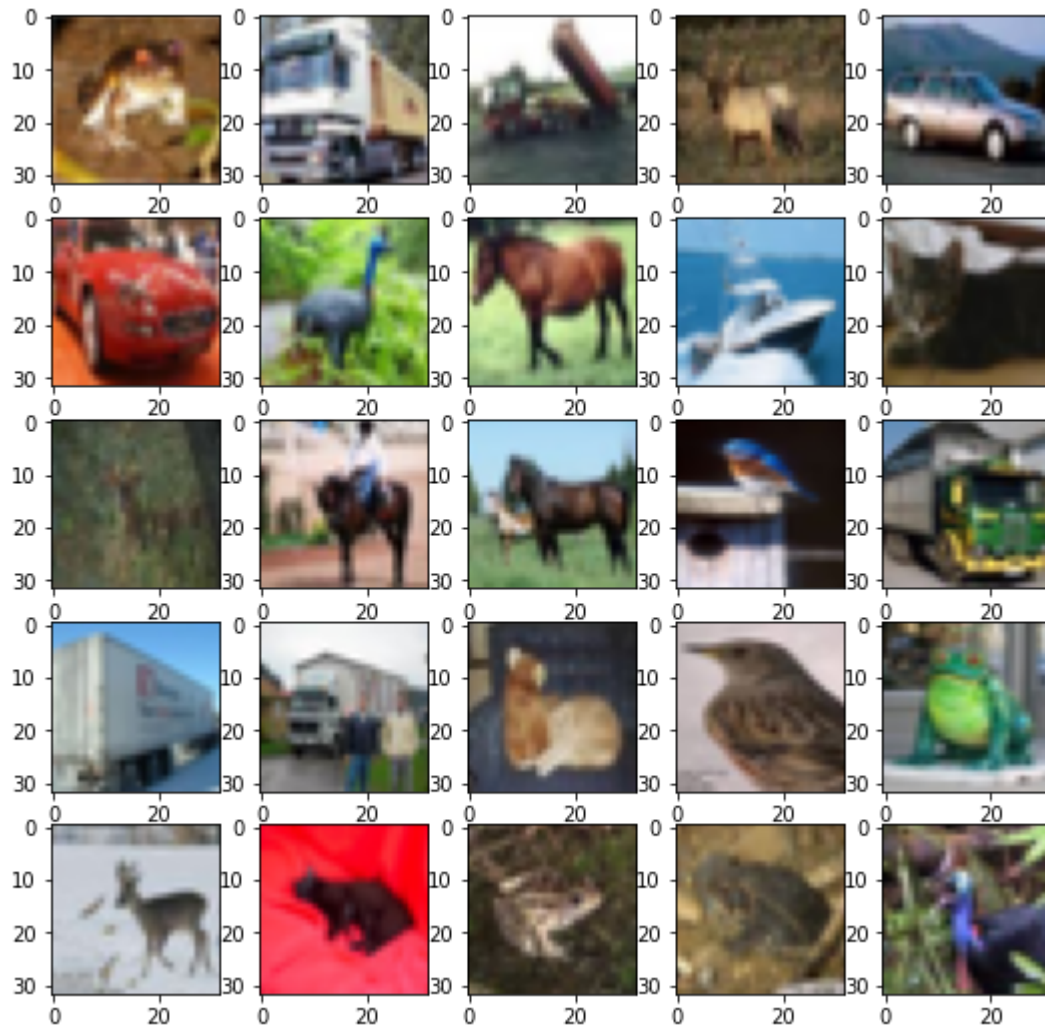
```
from keras.datasets import cifar10
from matplotlib import pyplot as plt
import numpy as np
from tensorflow.keras.utils import to_categorical
from keras.models import Sequential
from keras.utils import np_utils
from keras.layers import Dense,Activation,Dropout,LSTM,BatchNormalization
from keras.layers import Flatten
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.utils import load_img,img_to_array
from tensorflow.keras.models import load_model
```

```
from keras.datasets import cifar10
import matplotlib.pyplot as plt
from tensorflow.keras.utils import to_categorical
(x_train,y_train),(x_test,y_test) = cifar10.load_data()
x_train.shape
y_train.shape
```

```
(50000, 1)
```

```
from numpy import subtract
plt.figure(figsize=(9,9))
for i in range (25):
    plt.subplot(5,5,i+1)
    plt.imshow(x_train[i])
plt.show()
#0: máy bay
#1: xe hơi
#2: chim
#3: mèo
#4: hươu
#5: chó
#6: ếch
#7: ngựa
#8: tàu
... ..
```

```
#9: xe tai
```



```
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train/=255
x_test/=255
y_train=to_categorical(y_train)
y_test=to_categorical(y_test)
```

```
x_test.shape
```

```
x_test.shape
```

```
(10000, 32, 32, 3)
```

```
x_train.shape
```

```
(50000, 32, 32, 3)
```

```
y_test.shape
```

```
(10000, 10)
```

```
y_train.shape
```

```
(50000, 10)
```

```
model = Sequential()  
model.add(Flatten(input_shape=(32,32,3)))  
model.add(Dense(784,activation='relu'))  
model.add(Dense(784,activation='relu'))  
model.add(Dense(10,activation='Softmax'))  
model.summary()
```

```
Model: "sequential_1"
```

Layer (type)	Output Shape	Param #
flatten_1 (Flatten)	(None, 3072)	0
dense_3 (Dense)	(None, 784)	2409232
dense_4 (Dense)	(None, 784)	615440
dense_5 (Dense)	(None, 10)	7850

```
=====  
Total params: 3,032,522
```

```
Trainable params: 3,032,522
```

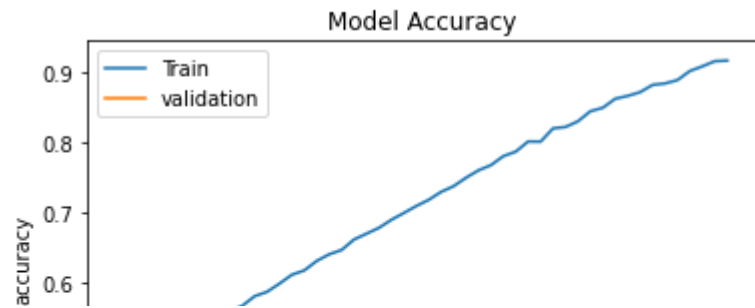
Non-trainable params: 0

```
opt=SGD(learning_rate=0.01,momentum=0.9)
model.compile(loss='categorical_crossentropy',optimizer=opt,metrics=['accuracy'])
history=model.fit(x_train,y_train,batch_size=128,epochs=50,verbose=1,validation_data=(x_test,y_test))
```

```
Epoch 1/50
391/391 [=====] - 21s 52ms/step - loss: 1.8176 - accuracy: 0.3503 - val_loss: 1.6735 - val_
Epoch 2/50
391/391 [=====] - 20s 50ms/step - loss: 1.6263 - accuracy: 0.4197 - val_loss: 1.6558 - val_
Epoch 3/50
391/391 [=====] - 20s 50ms/step - loss: 1.5462 - accuracy: 0.4499 - val_loss: 1.5198 - val_
Epoch 4/50
391/391 [=====] - 19s 50ms/step - loss: 1.4793 - accuracy: 0.4749 - val_loss: 1.4931 - val_
Epoch 5/50
391/391 [=====] - 26s 67ms/step - loss: 1.4264 - accuracy: 0.4949 - val_loss: 1.4633 - val_
Epoch 6/50
391/391 [=====] - 21s 54ms/step - loss: 1.3891 - accuracy: 0.5080 - val_loss: 1.4538 - val_
Epoch 7/50
391/391 [=====] - 19s 49ms/step - loss: 1.3598 - accuracy: 0.5165 - val_loss: 1.4015 - val_
Epoch 8/50
391/391 [=====] - 28s 73ms/step - loss: 1.3228 - accuracy: 0.5271 - val_loss: 1.4550 - val_
Epoch 9/50
391/391 [=====] - 21s 55ms/step - loss: 1.2807 - accuracy: 0.5469 - val_loss: 1.3699 - val_
Epoch 10/50
391/391 [=====] - 20s 51ms/step - loss: 1.2470 - accuracy: 0.5590 - val_loss: 1.3552 - val_
Epoch 11/50
391/391 [=====] - 19s 49ms/step - loss: 1.2231 - accuracy: 0.5662 - val_loss: 1.3504 - val_
Epoch 12/50
391/391 [=====] - 19s 49ms/step - loss: 1.1916 - accuracy: 0.5802 - val_loss: 1.3336 - val_
Epoch 13/50
391/391 [=====] - 20s 51ms/step - loss: 1.1680 - accuracy: 0.5865 - val_loss: 1.3562 - val_
Epoch 14/50
391/391 [=====] - 19s 49ms/step - loss: 1.1363 - accuracy: 0.5982 - val_loss: 1.3915 - val_
Epoch 15/50
391/391 [=====] - 20s 50ms/step - loss: 1.1035 - accuracy: 0.6110 - val_loss: 1.3188 - val_
Epoch 16/50
391/391 [=====] - 21s 53ms/step - loss: 1.0828 - accuracy: 0.6172 - val_loss: 1.3480 - val_
Epoch 17/50
```

```
391/391 [=====] - 19s 49ms/step - loss: 1.0460 - accuracy: 0.6308 - val_loss: 1.3178 - val_
Epoch 18/50
391/391 [=====] - 19s 49ms/step - loss: 1.0192 - accuracy: 0.6403 - val_loss: 1.3286 - val_
Epoch 19/50
391/391 [=====] - 19s 49ms/step - loss: 0.9944 - accuracy: 0.6465 - val_loss: 1.3558 - val_
Epoch 20/50
391/391 [=====] - 19s 49ms/step - loss: 0.9596 - accuracy: 0.6615 - val_loss: 1.3623 - val_
Epoch 21/50
391/391 [=====] - 20s 51ms/step - loss: 0.9342 - accuracy: 0.6699 - val_loss: 1.3268 - val_
Epoch 22/50
391/391 [=====] - 19s 49ms/step - loss: 0.9084 - accuracy: 0.6782 - val_loss: 1.3686 - val_
Epoch 23/50
391/391 [=====] - 19s 49ms/step - loss: 0.8813 - accuracy: 0.6898 - val_loss: 1.3928 - val_
Epoch 24/50
391/391 [=====] - 19s 49ms/step - loss: 0.8494 - accuracy: 0.6995 - val_loss: 1.3822 - val_
Epoch 25/50
391/391 [=====] - 19s 49ms/step - loss: 0.8291 - accuracy: 0.7095 - val_loss: 1.3708 - val_
Epoch 26/50
391/391 [=====] - 19s 49ms/step - loss: 0.7997 - accuracy: 0.7184 - val_loss: 1.4085 - val_
Epoch 27/50
391/391 [=====] - 19s 49ms/step - loss: 0.7684 - accuracy: 0.7298 - val_loss: 1.4488 - val_
Epoch 28/50
391/391 [=====] - 19s 49ms/step - loss: 0.7416 - accuracy: 0.7379 - val_loss: 1.4026 - val_
Epoch 29/50
```

```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['Train', 'validation'], loc='upper left')
plt.show()
```



```
model.save('/content/drive/MyDrive/BT AI/cifar10.h5')
```

```
| /
```

```
img = load_img('/content/drive/MyDrive/Anh AI/train/no/tau2.jpeg',target_size=(32,32))
```

```
plt.imshow(img)
```

```
img = img_to_array(img)
```

```
img = img.reshape(1,32,32,3)
```

```
img = img.astype('float32')
```

```
img = img/255
```

```
np.argmax(model.predict(img),axis=-1)
```

```
array([8])
```

