Trần Ngọc Đoàn - 19146175

https://github.com/DoanAI/Cifar100.git

```
import numpy as np
import matplotlib.pyplot as plt
from keras.layers import Dense, Activation, Dropout, BatchNormalization, LSTM
from keras.models import Sequential
from tensorflow.keras.utils import to_categorical
from keras import optimizers
from tensorflow.keras.optimizers import RMSprop
```

```
from skimage import color
from keras.datasets import cifar100
```

```
(x_train, y_train), (x_test, y_test) = cifar100.load_data()
```

```
x_train.shape
```

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

```
x = x_test
```

```
x_train  =color.rgb2gray(x_train)
print(x_train.shape)
x_test = color.rgb2gray(x_test)
print(x_test.shape)
```

```
    (50000, 32, 32)
    (10000, 32, 32)
```

```
x_train = x_train.reshape(50000,1024)
x_test = x_test.reshape(10000,1024)
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255
```

```
y_train = to_categorical(y_train,100)
y_test = to_categorical(y_test,100)
```

```
y_train.shape
```

```
    (50000, 100)
```

```
model = Sequential()
model.add(Dense(512,activation='relu',input_shape=(1024,)))
model.add(Dense(500,activation='relu'))
model.add(Dense(100,activation='relu'))
model.add(Dense(100,activation='softmax'))
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 dense (Dense)               (None, 512)               524800    
                                                                 
 dense_1 (Dense)             (None, 500)               256500    
                                                                 
 dense_2 (Dense)             (None, 100)               50100     
                                                                 
 dense_3 (Dense)             (None, 100)               10100     
                                                                 
=================================================================
Total params: 841,500
Trainable params: 841,500
Non-trainable params: 0
_________________________________________________________________
```

```
model.compile(loss='categorical_crossentropy',optimizer=RMSprop(),metrics=['accuracy']
```

```
history = model.fit(x_train,y_train,batch_size=128,epochs=100,verbose=1,validation_dat
```

```
391/391 [==============================] - 7s 17ms/step - loss: 3.2149 - accur
Epoch 73/100
391/391 [==============================] - 7s 17ms/step - loss: 3.2048 - accur
Epoch 74/100
391/391 [==============================] - 7s 17ms/step - loss: 3.1930 - accur
Epoch 75/100
391/391 [==============================] - 7s 17ms/step - loss: 3.1833 - accur
Epoch 76/100
391/391 [==============================] - 7s 17ms/step - loss: 3.1687 - accur
Epoch 77/100
391/391 [==============================] - 7s 17ms/step - loss: 3.1584 - accur
Epoch 78/100
391/391 [==============================] - 7s 17ms/step - loss: 3.1449 - accur
Epoch 79/100
391/391 [==============================] - 7s 17ms/step - loss: 3.1328 - accur
Epoch 80/100
391/391 [==============================] - 7s 17ms/step - loss: 3.1228 - accur
Epoch 81/100
391/391 [==============================] - 7s 18ms/step - loss: 3.1095 - accur
Epoch 82/100
391/391 [==============================] - 7s 17ms/step - loss: 3.0960 - accur
Epoch 83/100
391/391 [==============================] - 7s 17ms/step - loss: 3.0842 - accur
Epoch 84/100
391/391 [==============================] - 7s 17ms/step - loss: 3.0726 - accur
Epoch 85/100
```

```
391/391 [==============================] - 7s 17ms/step - loss: 3.0613 - accur
Epoch 86/100
391/391 [==============================] - 7s 17ms/step - loss: 3.0476 - accur
Epoch 87/100
391/391 [==============================] - 7s 17ms/step - loss: 3.0343 - accur
Epoch 88/100
391/391 [==============================] - 7s 17ms/step - loss: 3.0231 - accur
Epoch 89/100
391/391 [==============================] - 7s 17ms/step - loss: 3.0099 - accur
Epoch 90/100
391/391 [==============================] - 7s 17ms/step - loss: 3.0005 - accur
Epoch 91/100
391/391 [==============================] - 7s 17ms/step - loss: 2.9859 - accur
Epoch 92/100
391/391 [==============================] - 7s 17ms/step - loss: 2.9742 - accur
Epoch 93/100
391/391 [==============================] - 7s 17ms/step - loss: 2.9608 - accur
Epoch 94/100
391/391 [==============================] - 7s 17ms/step - loss: 2.9518 - accur
Epoch 95/100
391/391 [==============================] - 7s 18ms/step - loss: 2.9360 - accur
Epoch 96/100
391/391 [==============================] - 7s 17ms/step - loss: 2.9247 - accur
Epoch 97/100
391/391 [==============================] - 7s 18ms/step - loss: 2.9134 - accur
Epoch 98/100
391/391 [==============================] - 8s 19ms/step - loss: 2.8998 - accur
Epoch 99/100
391/391 [==============================] - 7s 17ms/step - loss: 2.8870 - accur
Epoch 100/100
391/391 [==============================] - 7s 17ms/step - loss: 2.8775 - accur
```

```
score = model.evaluate(x_test,y_test,verbose=0)
print(score[0])
print(score[1])
```

```
3.8961050510406494
0.16769999265670776
```

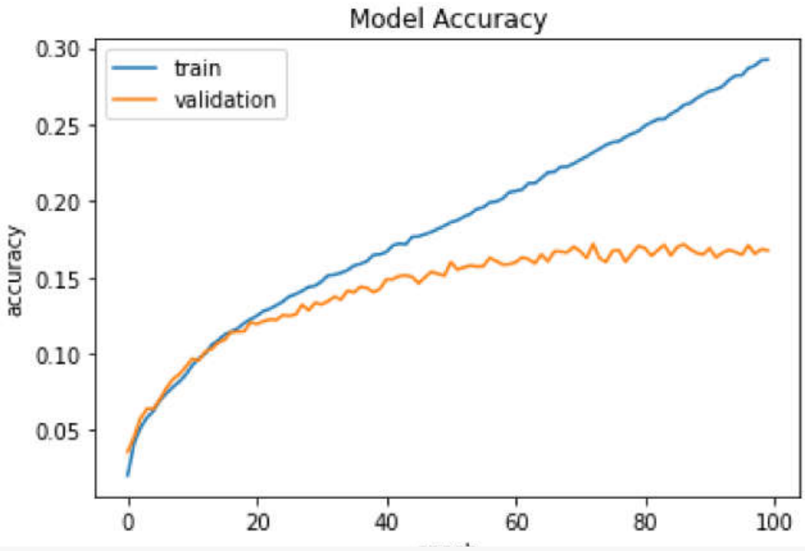
```
model.save('Cifar100.h5')
```

```
from keras.models import load_model
```

```
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')
```

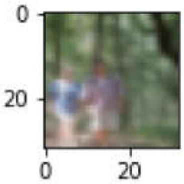
```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6: MatplotlibDeprec
        best
        upper right
        upper left
        lower left
        lower right
        right
        center left
        center right
        lower center
        upper center
        center
This will raise an exception in 3.3.

<matplotlib.legend.Legend at 0x7fe5abc9f990>
```

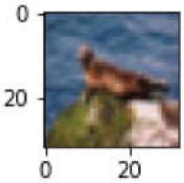


```
import numpy as np
y_pred = model.predict(x_test)
for i in range (9):
  plt.subplot(330+i+1)    # 330 mean: 3 hang 3 cot
  plt.imshow(x[i])
  plt.show()
  print(np.round(y_pred[i]))
```

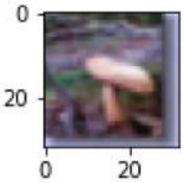
```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
```



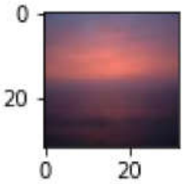
```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
```



```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
```

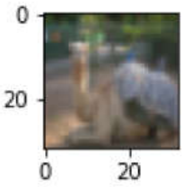


```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
```

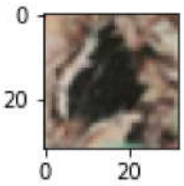


```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
```

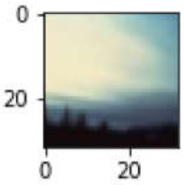
```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
```



```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
```



```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
```



```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
```

✓ 3 giây hoàn thành lúc 22:16