transfer learning

May 11, 2023

[2]:

```
import tensorflow
from tensorflow import keras
from keras import Sequential
from keras.layers import Dense,Flatten
from keras.applications.vgg16 import VGG16
```

[3]:

```
con_base=VGG16(
    weights='imagenet',
    include_top= False,
    input_shape=(150,150,3)
)
```

[4]:

```
con_base.summary()
```

```
Model: "vgg16"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 150, 150, 3)]     0

 block1_conv1 (Conv2D)       (None, 150, 150, 64)      1792

 block1_conv2 (Conv2D)       (None, 150, 150, 64)      36928

 block1_pool (MaxPooling2D)  (None, 75, 75, 64)        0

 block2_conv1 (Conv2D)       (None, 75, 75, 128)       73856

 block2_conv2 (Conv2D)       (None, 75, 75, 128)       147584

 block2_pool (MaxPooling2D)  (None, 37, 37, 128)       0

 block3_conv1 (Conv2D)       (None, 37, 37, 256)       295168

 block3_conv2 (Conv2D)       (None, 37, 37, 256)       590080

 block3_conv3 (Conv2D)       (None, 37, 37, 256)       590080
 block3_pool (MaxPooling2D)  (None, 18, 18, 256)       0

 block4_conv1 (Conv2D)       (None, 18, 18, 512)       1180160

 block4_conv2 (Conv2D)       (None, 18, 18, 512)       2359808

 block4_conv3 (Conv2D)       (None, 18, 18, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 9, 9, 512)         0

 block5_conv1 (Conv2D)       (None, 9, 9, 512)         2359808

 block5_conv2 (Conv2D)       (None, 9, 9, 512)         2359808

 block5_conv3 (Conv2D)       (None, 9, 9, 512)         2359808

 block5_pool (MaxPooling2D)  (None, 4, 4, 512)         0

=================================================================
Total params: 14,714,688
Trainable params: 14,714,688
Non-trainable params: 0
_________________________________________________________________
```

[5]:

```python
model=Sequential()
model.add(con_base)
model.add(Flatten())
model.add(Dense(256,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
```

[6]:

```python
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg16 (Functional)          (None, 4, 4, 512)         14714688

 flatten (Flatten)           (None, 8192)              0

 dense (Dense)               (None, 256)               2097408

 dense_1 (Dense)             (None, 1)                 257

=================================================================
Total params: 16,812,353
Trainable params: 16,812,353
Non-trainable params: 0
```

```
-----------------------------------------------------------------
```

[7]:

```
con_base.trainable=False
```

[8]:

```
# generators
tains_ds = keras.utils.image_dataset_from_directory(
     directory = r'C:\Users\Rakesh\Downloads\archive (2)\train',
    labels='inferred',
    label_mode = 'int',
    batch_size=32,
    image_size=(150,150)
)
validation_ds = keras.utils.image_dataset_from_directory(
    directory =r'C:\Users\Rakesh\Downloads\archive (2)\test',
    labels='inferred',
    label_mode = 'int',
    batch_size=32,
    image_size=(150,150)
)
```

```
Found 20000 files belonging to 2 classes.
Found 5000 files belonging to 2 classes.
```

[9]:

```
def normalize(image,label):
    image=tensorflow.cast(image/255. ,tensorflow.float32)
    return image,label
x=tains_ds.map(normalize)
y=validation_ds.map(normalize)
```

[10]:

```
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
history=model.fit(x,epochs=4,validation_data=y)
```

```
Epoch 1/4
625/625 [==============================] - 4557s 7s/step - loss: 0.2810 -
accuracy: 0.8792 - val_loss: 0.2262 - val_accuracy: 0.9080
Epoch 2/4
625/625 [==============================] - 4782s 8s/step - loss: 0.1844 -
accuracy: 0.9247 - val_loss: 0.2122 - val_accuracy: 0.9130
Epoch 3/4
625/625 [==============================] - 2961s 5s/step - loss: 0.1620 -
accuracy: 0.9335 - val_loss: 0.2274 - val_accuracy: 0.9102
Epoch 4/4
625/625 [==============================] - 3090s 5s/step - loss: 0.1323 -
accuracy: 0.9464 - val_loss: 0.2151 - val_accuracy: 0.9176
```

[12]:

```
import joblib
joblib.dump(history, 'catdogmodel')
```

```
Keras weights file (<HDF5 file "variables.h5" (mode r+)>) saving:
```

```
…layers\dense
…vars
…0
…1
…layers\dense_1
…vars
…0
…1
…layers\flatten
…vars
…layers\functional
…vars
…layers\functional\layers\conv2d
…vars
…0
…1
…layers\functional\layers\conv2d_1
…vars
…0
…1
…layers\functional\layers\conv2d_10
…vars
…0
…1
…layers\functional\layers\conv2d_11
…vars
…0
…1
…layers\functional\layers\conv2d_12
…vars
…0
…1
…layers\functional\layers\conv2d_2
…vars
…0
…1
…layers\functional\layers\conv2d_3
…vars
…0
…1
…layers\functional\layers\conv2d_4
…vars
…0
…1
…layers\functional\layers\conv2d_5
…vars
…0
…1
```

```
…layers\functional\layers\conv2d_6
…vars
…0
…1
…layers\functional\layers\conv2d_7
…vars
…0
…1
…layers\functional\layers\conv2d_8
…vars
…0
…1
…layers\functional\layers\conv2d_9
…vars
…0
…1
…layers\functional\layers\input_layer
…vars
…layers\functional\layers\max_pooling2d
…vars
…layers\functional\layers\max_pooling2d_1
…vars
…layers\functional\layers\max_pooling2d_2
…vars
…layers\functional\layers\max_pooling2d_3
…vars
…layers\functional\layers\max_pooling2d_4
…vars
…metrics\mean
…vars
…0
…1
…metrics\mean_metric_wrapper
…vars
…0
…1
…optimizer
…vars
…0
…1
…2
…3
…4
…5
…6
…7
…8
…vars
```

```
Keras model archive saving:
File Name                                             Modified              Size
config.json                                    2023-03-30 05:19:44         11542
metadata.json                                  2023-03-30 05:19:44            64
variables.h5                                   2023-03-30 05:19:45      84099632
```

[12]:

```
['catdogmodel']
```

[3]:

```python
import joblib
```

[4]:

```python
catdogmodel = joblib.load('catdogmodel')
```

```
Keras model archive loading:
File Name                                             Modified              Size
config.json                                    2023-03-30 05:19:44         11542
metadata.json                                  2023-03-30 05:19:44            64
variables.h5                                   2023-03-30 05:19:44      84099632
Keras weights file (<HDF5 file "variables.h5" (mode r)>) loading:
…layers\dense
…vars
…0
…1
…layers\dense_1
…vars
…0
…1
…layers\flatten
…vars
…layers\functional
…vars
…layers\functional\layers\conv2d
…vars
…0
…1
…layers\functional\layers\conv2d_1
…vars
…0
…1
…layers\functional\layers\conv2d_10
…vars
…0
…1
…layers\functional\layers\conv2d_11
…vars
…0
…1
…layers\functional\layers\conv2d_12
…vars
```

```
…0
…1
…layers\functional\layers\conv2d_2
…vars
…0
…1
…layers\functional\layers\conv2d_3
…vars
…0
…1
…layers\functional\layers\conv2d_4
…vars
…0
…1
…layers\functional\layers\conv2d_5
…vars
…0
…1
…layers\functional\layers\conv2d_6
…vars
…0
…1
…layers\functional\layers\conv2d_7
…vars
…0
…1
…layers\functional\layers\conv2d_8
…vars
…0
…1
…layers\functional\layers\conv2d_9
…vars
…0
…1
…layers\functional\layers\input_layer
…vars
…layers\functional\layers\max_pooling2d
…vars
…layers\functional\layers\max_pooling2d_1
…vars
…layers\functional\layers\max_pooling2d_2
…vars
…layers\functional\layers\max_pooling2d_3
…vars
…layers\functional\layers\max_pooling2d_4
…vars
…metrics\mean
…vars
```

```
…0
…1
…metrics\mean_metric_wrapper
…vars
…0
…1
…optimizer
…vars
…0
…1
…2
…3
…4
…5
…6
…7
…8
…vars
```

[ ]:

```
catdogmodel.predict([[]])
```

[1]:

```
import matplotlib.pyplot as plt

plt.plot(history.history['accuracy'],color='red',label='train')
plt.plot(history.history['val_accuracy'],color='blue',label='validation')
plt.legend()
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [1], in <cell line: 3>()
      1 import matplotlib.pyplot as plt
----> 3 plt.plot(history.history['accuracy'],color='red',label='train')
      4 plt.plot(history.history['val_accuracy'],color='blue',label='validation
      5 plt.legend()

NameError: name 'history' is not defined
```

[ ]:

```
plt.plot(history.history['loss'],color='red',label='train')
plt.plot(history.history['val_loss'],color='blue',label='validation')
plt.legend()
plt.show()
```