In [1]:

```
!pip install --upgrade keras_applications==1.0.6 keras==2.2.4

import os
if os.path.exists('./data') == False:
    from modelarts.session import Session
    session = Session()

    session.download_data(
    bucket_path="modelarts-labs/notebook/DL_image_recognition/image_recognition.tar.gz",
    path="./image_recognition.tar.gz")

    # 使用tar命令解压资源包
    !tar xf ./image_recognition.tar.gz

    # 清理压缩包
    !rm -f ./image_recognition.tar.gz
```

```
Requirement already up-to-date: keras_applications==1.0.6 in /opt/conda/envs/python3
6_tf/lib/python3.6/site-packages
Requirement already up-to-date: keras==2.2.4 in /opt/conda/envs/python36_tf/lib/pyth
on3.6/site-packages
Requirement already up-to-date: h5py in /opt/conda/envs/python36_tf/lib/python3.6/si
te-packages (from keras_applications==1.0.6)
Requirement already up-to-date: numpy>=1.9.1 in /opt/conda/envs/python36_tf/lib/pyth
on3.6/site-packages (from keras_applications==1.0.6)
Requirement already up-to-date: pyyaml in /opt/conda/envs/python36_tf/lib/python3.6/
site-packages (from keras==2.2.4)
Requirement already up-to-date: keras-preprocessing>=1.0.5 in /opt/conda/envs/python
36_tf/lib/python3.6/site-packages (from keras==2.2.4)
Requirement already up-to-date: six>=1.9.0 in /home/jovyan/modelarts-sdk (from keras
==2.2.4)
Requirement already up-to-date: scipy>=0.14 in /opt/conda/envs/python36_tf/lib/pytho
n3.6/site-packages (from keras==2.2.4)
You are using pip version 9.0.1, however version 19.1.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

In [2]:

```
from keras.applications.vgg16 import VGG16
from keras.preprocessing import image
import numpy as np

from keras.preprocessing import image
from keras.models import Model
from keras.layers import Dense, GlobalAveragePooling2D
from keras import backend as K
from keras.models import load_model

from keras.preprocessing.image import ImageDataGenerator
```

```
Using TensorFlow backend.
```

In [3]:

```
import os
from PIL import Image
def load_data():
    dirname = "./data"
    path = "./data"

    num_train_samples = 25000

    x_train = np.empty((num_train_samples, 224,224,3), dtype='uint8')
    y_train = np.empty((num_train_samples,1), dtype='uint8')
    index = 0
    for file in os.listdir("./data"):
        image = Image.open(os.path.join(dirname,file)).resize((224,224))
        image = np.array(image)
        x_train[index,:,:,:] = image

        if "cat" in file:
            y_train[index,0] =1
        elif "dog" in file:
            y_train[index,0] =0

        index += 1
    return (x_train, y_train)
```

In [4]:

```
(x_train, y_train) = load_data()
```

In [5]:

```
print(x_train.shape)
print(y_train.shape)
```

```
(25000, 224, 224, 3)
(25000, 1)
```

In [6]:

```
from keras.utils import np_utils
def process_data(x_train,y_train):
    x_train = x_train.astype(np.float32)
    x_train /= 255
    n_classes = 2
    y_train = np_utils.to_categorical(y_train, n_classes)
    return x_train,y_train
```

In [7]:

```
def build_model(base_model):
    x = base_model.output
    x = GlobalAveragePooling2D()(x)
    predictions = Dense(2, activation='softmax')(x)
    model = Model(inputs=base_model.input, outputs=predictions)
    print(type(model))
    return model
```

In [8]:

```
x_train,y_train= process_data(x_train,y_train)
print(x_train.shape)
print(y_train.shape)
```

```
(25000, 224, 224, 3)
(25000, 2)
```

In [9]:

```
base_model = VGG16(weights='imagenet', include_top=False)
```

```
Downloading data from https://github.com/fchollet/deep-learning-models/releases/down
load/v0.1/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5 (https://github.com/fcho
llet/deep-learning-models/releases/download/v0.1/vgg16_weights_tf_dim_ordering_tf_ke
rnels_notop.h5)
58892288/58889256 [==============================] - 222s 4us/step
```

In [10]:

```
for layer in base_model.layers:
    layer.trainable = False

model = build_model(base_model)
model.summary()
```

<class 'keras.engine.training.Model'>

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_1 (InputLayer) | (None, None, None, 3) | 0 |
| block1_conv1 (Conv2D) | (None, None, None, 64) | 1792 |
| block1_conv2 (Conv2D) | (None, None, None, 64) | 36928 |
| block1_pool (MaxPooling2D) | (None, None, None, 64) | 0 |
| block2_conv1 (Conv2D) | (None, None, None, 128) | 73856 |
| block2_conv2 (Conv2D) | (None, None, None, 128) | 147584 |
| block2_pool (MaxPooling2D) | (None, None, None, 128) | 0 |
| block3_conv1 (Conv2D) | (None, None, None, 256) | 295168 |
| block3_conv2 (Conv2D) | (None, None, None, 256) | 590080 |
| block3_conv3 (Conv2D) | (None, None, None, 256) | 590080 |
| block3_pool (MaxPooling2D) | (None, None, None, 256) | 0 |
| block4_conv1 (Conv2D) | (None, None, None, 512) | 1180160 |
| block4_conv2 (Conv2D) | (None, None, None, 512) | 2359808 |
| block4_conv3 (Conv2D) | (None, None, None, 512) | 2359808 |
| block4_pool (MaxPooling2D) | (None, None, None, 512) | 0 |
| block5_conv1 (Conv2D) | (None, None, None, 512) | 2359808 |
| block5_conv2 (Conv2D) | (None, None, None, 512) | 2359808 |
| block5_conv3 (Conv2D) | (None, None, None, 512) | 2359808 |
| block5_pool (MaxPooling2D) | (None, None, None, 512) | 0 |
| global_average_pooling2d_1 ( | (None, 512) | 0 |
| dense_1 (Dense) | (None, 2) | 1026 |

Total params: 14,715,714
Trainable params: 1,026
Non-trainable params: 14,714,688

In [11]:

```
import keras
opt = keras.optimizers.rmsprop(lr=0.0001, decay=1e-6)
model.compile(loss='binary_crossentropy',
              optimizer=opt,
              metrics=['accuracy'])
```

In [12]:

```
from keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau
es = EarlyStopping(monitor='val_acc', baseline=0.9, patience=15, verbose=1, mode='auto')
cp = ModelCheckpoint(filepath="./ckpt_vgg16_dog_and_cat.h5", monitor="val_acc", verbose=1, save_best
lr = ReduceLROnPlateau(monitor="val_acc", factor=0.1, patience=10, verbose=1, mode="auto", min_lr=0)
callbacks = [es,cp,lr]
```

In [13]:

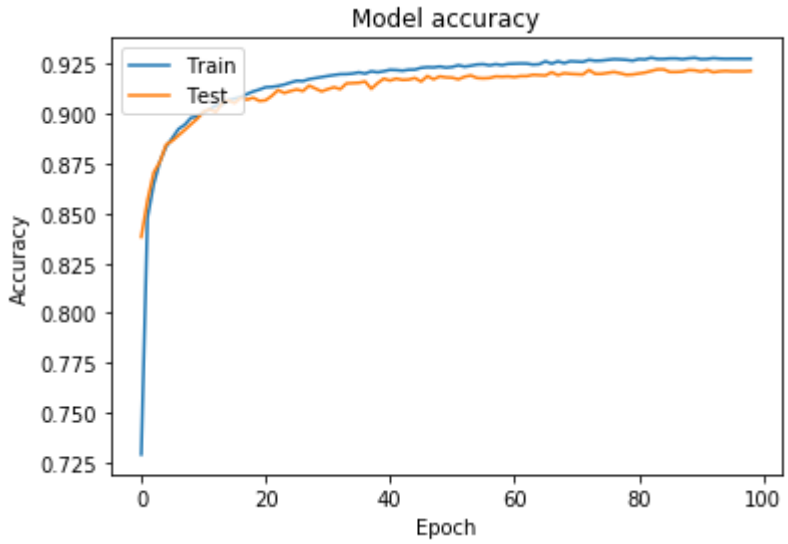
```
history = model.fit(x=x_train,
                    y=y_train,
                    batch_size=16,
                    epochs=100,
                    verbose=1,
                    callbacks=callbacks,
                    validation_split=0.25,
                    shuffle=True,
                    initial_epoch=0)
```

```
0.9277 - val_loss: 0.1932 - val_acc: 0.9214

Epoch 00096: val_acc did not improve from 0.92240
Epoch 97/100
18750/18750 [==============================] - 71s 4ms/step - loss: 0.1800 - acc:
0.9277 - val_loss: 0.1932 - val_acc: 0.9214

Epoch 00097: val_acc did not improve from 0.92240
Epoch 98/100
18750/18750 [==============================] - 71s 4ms/step - loss: 0.1800 - acc:
0.9277 - val_loss: 0.1932 - val_acc: 0.9214

Epoch 00098: val_acc did not improve from 0.92240
Epoch 99/100
18750/18750 [==============================] - 71s 4ms/step - loss: 0.1800 - acc:
0.9277 - val_loss: 0.1932 - val_acc: 0.9216

Epoch 00099: val_acc did not improve from 0.92240
Epoch 00099: early stopping
```

In [15]:

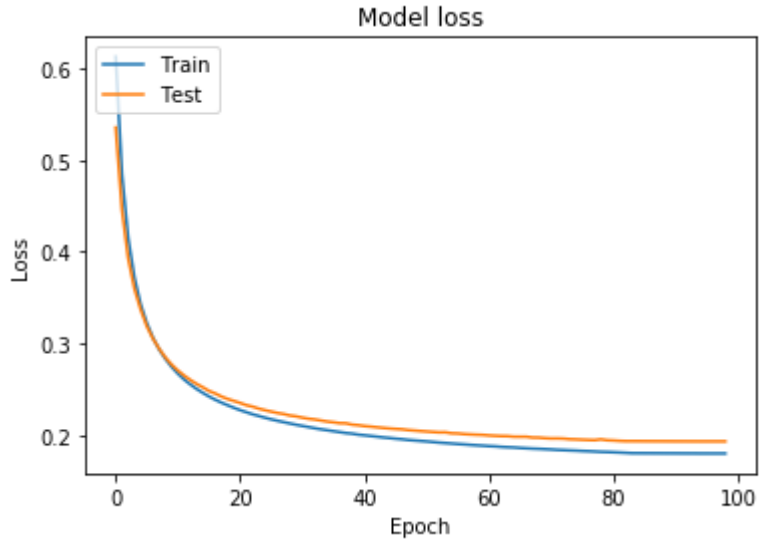
```
import matplotlib.pyplot as plt
print(history.history['val_acc'][0])
# 绘制训练 & 验证的准确率值
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```

0.8382400000190735




In [16]:

```
# 绘制训练 & 验证的损失值
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```



In [17]:

```
base_model = VGG16(weights='imagenet', include_top=False)
model = build_model(base_model)
model.compile(loss='binary_crossentropy',
              optimizer=opt,
              metrics=['accuracy'])
model.summary()
```

```
<class 'keras.engine.training.Model'>
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         (None, None, None, 3)     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, None, None, 64)    1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, None, None, 64)    36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, None, None, 64)    0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, None, None, 128)   73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, None, None, 128)   147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, None, None, 128)   0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, None, None, 256)   295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, None, None, 256)   590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, None, None, 256)   590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, None, None, 256)   0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, None, None, 512)   1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, None, None, 512)   2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, None, None, 512)   2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, None, None, 512)   0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, None, None, 512)   2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, None, None, 512)   2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, None, None, 512)   2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, None, None, 512)   0
_________________________________________________________________
global_average_pooling2d_2 ( (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 1026
=================================================================
Total params: 14,715,714
Trainable params: 14,715,714
Non-trainable params: 0
_________________________________________________________________
```

In [18]:

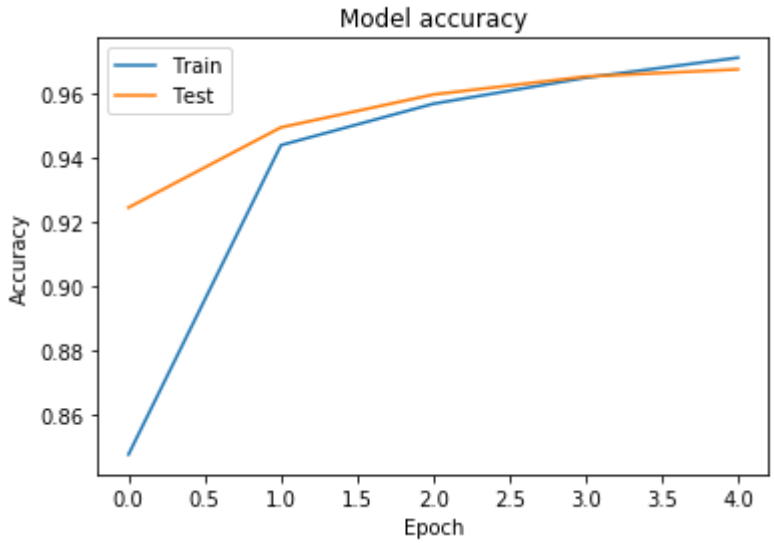
```
history = model.fit(x=x_train,
                    y=y_train,
                    batch_size=16,
                    epochs=5,
                    verbose=1,
                    callbacks=callbacks,
                    validation_split=0.25,
                    shuffle=True,
                    initial_epoch=0)
```

```
Train on 18750 samples, validate on 6250 samples
Epoch 1/5
18750/18750 [==============================] - 168s 9ms/step - loss: 0.3699 - acc:
0.8477 - val_loss: 0.1915 - val_acc: 0.9246

Epoch 00001: val_acc improved from 0.92240 to 0.92464, saving model to ./ckpt_vgg16_
dog_and_cat.h5
Epoch 2/5
18750/18750 [==============================] - 166s 9ms/step - loss: 0.1488 - acc:
0.9441 - val_loss: 0.1298 - val_acc: 0.9496

Epoch 00002: val_acc improved from 0.92464 to 0.94960, saving model to ./ckpt_vgg16_
dog_and_cat.h5
Epoch 3/5
18750/18750 [==============================] - 164s 9ms/step - loss: 0.1103 - acc:
0.9570 - val_loss: 0.1028 - val_acc: 0.9598

Epoch 00003: val_acc improved from 0.94960 to 0.95984, saving model to ./ckpt_vgg16_
dog_and_cat.h5
Epoch 4/5
18750/18750 [==============================] - 164s 9ms/step - loss: 0.0910 - acc:
0.9651 - val_loss: 0.0892 - val_acc: 0.9654

Epoch 00004: val_acc improved from 0.95984 to 0.96544, saving model to ./ckpt_vgg16_
dog_and_cat.h5
Epoch 5/5
18750/18750 [==============================] - 164s 9ms/step - loss: 0.0778 - acc:
0.9713 - val_loss: 0.0850 - val_acc: 0.9677

Epoch 00005: val_acc improved from 0.96544 to 0.96768, saving model to ./ckpt_vgg16_
dog_and_cat.h5
```

In [19]:

```
import matplotlib.pyplot as plt
print(history.history['val_acc'][0])
# 绘制训练 & 验证的准确率值
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show();
```

0.92464




In [ ]: