







In [2]:

```

from keras.preprocessing.image import ImageDataGenerator
from glob import glob

ROWS = 224
COLS = 224

if os.path.exists('./data-for-gen') == False:
    !mkdir -p data-for-gen/train/dog
    !mkdir -p data-for-gen/train/cat
    !mkdir -p data-for-gen/val/dog
    !mkdir -p data-for-gen/val/cat

DATA_DIR = './data-for-gen/' # 数据集路径
dog_glob = glob('./data/dog*.jpg')
cat_glob = glob('./data/cat*.jpg')

# 我们使用25%的数据作为验证集:
val_split = 0.25

index = int(len(dog_glob) * val_split)

def gen_lnk_cmds(class_name, class_glob):
    cmds = ''
    for i in range(len(class_glob)):
        filename = os.path.basename(class_glob[i])

        src_path = os.path.realpath(class_glob[i])
        sample_type = 'train' if i > index else 'val'
        lnk_path = os.path.realpath('./data-for-gen/{}/{}/{}'.format(sample_type, class_name, filename))
        cmds = cmds + 'ln -s {} {}\n'.format(src_path, lnk_path)

    return cmds

# 准备dog图片
print('prepare dog images for data augmentation')
with open('./tmp_gen_dogs.sh', 'w') as f:
    link_cmds = gen_lnk_cmds('dog', dog_glob)
    f.write(link_cmds)
    !sh ./tmp_gen_dogs.sh
    !rm ./tmp_gen_dogs.sh

# 准备cat图片
print('prepare cat images for data augmentation')
with open('./tmp_gen_cats.sh', 'w') as f:
    link_cmds = gen_lnk_cmds('cat', cat_glob)
    f.write(link_cmds)
    !sh ./tmp_gen_cats.sh
    !rm ./tmp_gen_cats.sh

```

Using TensorFlow backend.

```

prepare dog images for data augmentation
prepare cat images for data augmentation

```

In [3]:

```
from PIL import Image
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline

train_datagen = temp_val_datagen = ImageDataGenerator(
    rescale=1.0/255, # ImageDataGenerator使用[0-1]表示RGB色值, 加入rescale以正常显示图片
    rotation_range=10, # 旋转角度范围, 以角度为单位
    width_shift_range=0.2, # 宽度的偏移范围
    height_shift_range=0.2, # 高度的偏移范围
    shear_range=0.1, # 图片裁剪范围
    zoom_range=0.1, # 图片缩放范围
    fill_mode='nearest', # 图片处理后的填充模式
    horizontal_flip=True) # 是否随机水平翻转图片

val_datagen = temp_val_datagen = ImageDataGenerator(
    rescale=1.0/255,
    rotation_range=10,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.1,
    zoom_range=0.1,
    fill_mode='nearest',
    horizontal_flip=True)
```

In [4]:

```

temp_val_datagen = ImageDataGenerator(
    rescale=1.0/255,
    rotation_range=10,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.1,
    zoom_range=0.1,
    fill_mode='nearest',
    horizontal_flip=True)

from PIL import Image
sample_img = Image.open(glob('./data/dog*.jpg')[0])
sample_img = np.expand_dims(sample_img, axis=0)

i = 0
plt.figure(figsize=(20, 10))
for batch in temp_val_datagen.flow(sample_img):
    plt.subplot(2, 4, i+1)
    plt.axis('off')
    plt.imshow(np.squeeze(batch))

    i += 1
    if i >= 8:
        break

plt.show()

```



In [5]:

```

train_generator = train_datagen.flow_from_directory('./data-for-gen/train',
                                                    target_size=(ROWS, COLS), batch_size=16, class_mode='binary_crossentropy')

val_generator = val_datagen.flow_from_directory('./data-for-gen/val',
                                                target_size=(ROWS, COLS), batch_size=16, class_mode='binary_crossentropy')

```

Found 18748 images belonging to 2 classes.

Found 6252 images belonging to 2 classes.

In [6]:

```

from keras.applications.vgg16 import VGG16
from keras.models import Model
from keras.layers import Flatten, Dense, GlobalAveragePooling2D
from keras.optimizers import RMSprop

optimizer = RMSprop(lr=1e-4, decay=1e-6) # 优化器使用RMSprop, 设置学习率是1e-4
objective = 'binary_crossentropy' # loss 函数使用交叉熵

base_model = VGG16(weights=None, include_top=False, input_shape=(COLS, ROWS, 3))

x = base_model.output
x = GlobalAveragePooling2D()(x)

output = Dense(1, activation='sigmoid')(x)

model = Model(input=base_model.input, output=output)
model.summary()

model.compile(loss=objective, optimizer=optimizer, metrics=['accuracy'])

```

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808

block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
global_average_pooling2d_1	(None, 512)	0
dense_1 (Dense)	(None, 1)	513
=====		
Total params: 14,715,201		
Trainable params: 14,715,201		
Non-trainable params: 0		

```
/opt/conda/envs/python36_tf/lib/python3.6/site-packages/ipykernel_launcher.py:16: UserWarning: Update your `Model` call to the Keras 2 API: `Model(inputs=Tensor("in...", outputs=Tensor("de..."))`
  app.launch_new_instance()
```

In [7]:

```
epochs = 50 # 训练轮数

from keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLRonPlateau

# early stopping策略
early_stopping = EarlyStopping(monitor='val_loss', patience=20, verbose=1, mode='auto')
mcp = ModelCheckpoint('aug_weights_{epoch:03d}_{acc:.4f}_{val_acc:.4f}.h5',
                      monitor='val_acc', save_best_only=True, save_weights_only=True, verbose=1)

reduce_lr = ReduceLRonPlateau(monitor='val_acc', factor=0.2, patience=5, min_lr=1e-9, verbose=1)
# 开始训练
hist = model.fit_generator(
    epochs=epochs,
    generator=train_generator,
    steps_per_epoch=len(train_generator),
    validation_data=val_generator,
    validation_steps=len(val_generator),
    shuffle=True,
    callbacks=[early_stopping, mcp, reduce_lr])
```

```
Epoch 00047: val_acc did not improve from 0.94226
Epoch 48/50
1172/1172 [=====] - 279s 238ms/step - loss: 0.1492 - acc:
0.9504 - val_loss: 0.2097 - val_acc: 0.9315

Epoch 00048: val_acc did not improve from 0.94226

Epoch 00048: ReduceLRonPlateau reducing learning rate to 1.600000018697756e-07.
Epoch 49/50
1172/1172 [=====] - 278s 237ms/step - loss: 0.1497 - acc:
0.9490 - val_loss: 0.1879 - val_acc: 0.9370

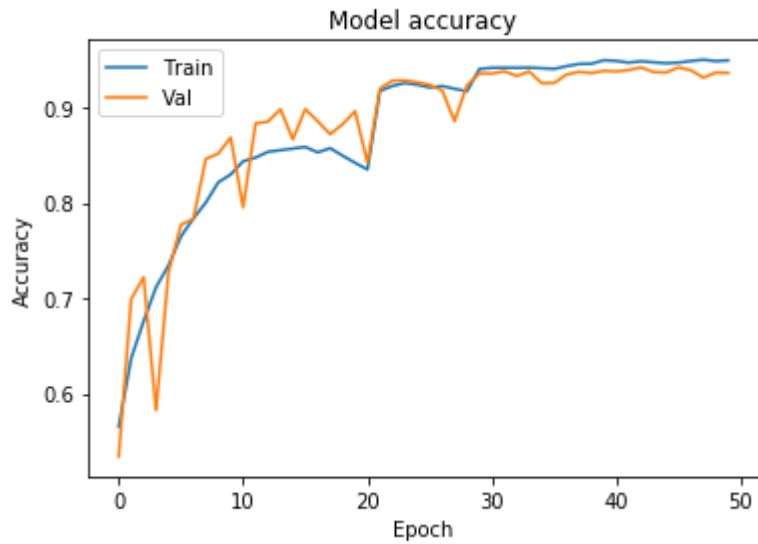
Epoch 00049: val_acc did not improve from 0.94226
Epoch 50/50
1172/1172 [=====] - 279s 238ms/step - loss: 0.1480 - acc:
0.9497 - val_loss: 0.1867 - val_acc: 0.9365

Epoch 00050: val_acc did not improve from 0.94226
Epoch 00050: early stopping
```



In [8]:

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



In [ ]: