In [1]:

```
import numpy as np
from keras.models import load_model
from keras.preprocessing import image
from keras.preprocessing.image import ImageDataGenerator
```

```
Using TensorFlow backend.
```

In [2]:

```
def test_image():
    img_path = input("Enter your image name in validate folder (with extensio
n):")
    img_path = "validate/" + img_path

    img = image.load_img(img_path, target_size=(150, 150))
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)

    model = load_model('model_min.h5')
    preds = model.predict(x)
    # output = np.array(preds, np.int32)

    print(preds)
    if preds == 1:
        print("Dog!!!")
    else:
        print("Cat!!!")
```

In [3]:

```
def fine_tune_on_all_samples():
    train_data_dir = 'cats_and_dogs_medium/train'  # Path to training images
    validation_data_dir = 'cats_and_dogs_medium/test'  # Validation and test s
et are the same here

    nb_train_samples = 1500
    nb_validation_samples = 45
    epochs = 2
    batch_size = 16
    img_width, img_height = 150, 150

    # Prepare data augmentation configuration
    train_datagen = ImageDataGenerator(
        rescale=1. / 255,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True)

    test_datagen = ImageDataGenerator(rescale=1. / 255)

    train_generator = train_datagen.flow_from_directory(
        train_data_dir,
        target_size=(img_height, img_width),
        batch_size=batch_size,
        class_mode='binary')

    validation_generator = test_datagen.flow_from_directory(
        validation_data_dir,
        target_size=(img_height, img_width),
        batch_size=batch_size,
        class_mode='binary')

    model = load_model('model_min.h5')
        # Fine-tune the model
    model.fit_generator(
        train_generator,
        samples_per_epoch=nb_train_samples // batch_size,
        epochs=epochs,
        validation_data=validation_generator,
        validation_steps=nb_validation_samples // batch_size)
```

In [4]:
```
# fine_tune with saved model
fine_tune_on_all_samples()
```

```
Found 7178 images belonging to 2 classes.
Found 1796 images belonging to 2 classes.
WARNING:tensorflow:From C:\Users\abbaz\Anaconda\envs\cecs551\lib\site-package
s\tensorflow\python\framework\op_def_library.py:263: colocate_with (from tens
orflow.python.framework.ops) is deprecated and will be removed in a future ve
rsion.
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From C:\Users\abbaz\Anaconda\envs\cecs551\lib\site-package
s\tensorflow\python\ops\math_ops.py:3066: to_int32 (from tensorflow.python.op
s.math_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
```

```
C:\Users\abbaz\Anaconda\envs\cecs551\lib\site-packages\ipykernel_launcher.py:
39: UserWarning: Update your `fit_generator` call to the Keras 2 API: `fit_ge
nerator(<keras_pre..., epochs=2, validation_data=<keras_pre..., validation_st
eps=2, steps_per_epoch=5)`
```

```
Epoch 1/2
5/5 [==============================] - 28s 6s/step - loss: 0.3496 - acc: 0.87
50 - val_loss: 0.1003 - val_acc: 0.9688
Epoch 2/2
5/5 [==============================] - 27s 5s/step - loss: 0.2126 - acc: 0.91
25 - val_loss: 0.2758 - val_acc: 0.9375
```

In [ ]:
```

```