Binary_Image_Classification

October 21, 2018

```
In [1]: import tensorflow as tf
        import os, shutil
        from PIL import Image
        import matplotlib.pyplot as plt
        import warnings
        warnings.filterwarnings('ignore')
```

```
In [2]: !ls
```

```
666.jpg                          Untitled.ipynb           mnistTF.ipynb
Conv_Net_Dogs_Cats.ipynb  Untitled1.ipynb           mnistTF.model
Dogs_and_Cats_Data.ipynb  X_train.pickle           requirements.txt
Junk                          cats_and_dogs_small.old  y_train.pickle
PetImages                  core
```

```
In [3]: !mkdir cats_and_dogs_small
```

```
In [4]: !ls
```

```
666.jpg                          Untitled.ipynb           core
Conv_Net_Dogs_Cats.ipynb  Untitled1.ipynb           mnistTF.ipynb
Dogs_and_Cats_Data.ipynb  X_train.pickle           mnistTF.model
Junk                          cats_and_dogs_small           requirements.txt
PetImages                  cats_and_dogs_small.old  y_train.pickle
```

```
In [5]: source_cats = 'PetImages/Cat'
        source_dogs = 'PetImages/Dog'

        base_dir = 'cats_and_dogs_small'
```

```
In [6]: train_dir = os.path.join(base_dir, 'train')
        os.mkdir(train_dir)

        validation_dir = os.path.join(base_dir, 'validation')
        os.mkdir(validation_dir)

        test_dir = os.path.join(base_dir, 'test')
        os.mkdir(test_dir)
```

```
In [7]: !ls cats_and_dogs_small/
```

```
test  train  validation
```

```
In [8]: train_cats_dir = os.path.join(train_dir, 'cats')
        os.mkdir(train_cats_dir)

        train_dogs_dir = os.path.join(train_dir, 'dogs')
        os.mkdir(train_dogs_dir)

        validation_cats_dir = os.path.join(validation_dir, 'cats')
        os.mkdir(validation_cats_dir)

        validation_dogs_dir = os.path.join(validation_dir, 'dogs')
        os.mkdir(validation_dogs_dir)

        test_cats_dir = os.path.join(test_dir, 'cats')
        os.mkdir(test_cats_dir)

        test_dogs_dir = os.path.join(test_dir, 'dogs')
        os.mkdir(test_dogs_dir)
```

```
In [9]: !ls cats_and_dogs_small/train
```

```
cats  dogs
```

```
In [10]: !ls cats_and_dogs_small/validation
```

```
cats  dogs
```

```
In [11]: !ls cats_and_dogs_small/test
```

```
cats  dogs
```

```
In [12]: fnames = ['{}.jpg'.format(i) for i in range(1000)]
         for fname in fnames:
             src = os.path.join(source_cats, fname)
             dst = os.path.join(train_cats_dir, fname)
             shutil.copyfile(src, dst)

         fnames = ['{}.jpg'.format(i) for i in range(1000, 1500)]
         for fname in fnames:
             src = os.path.join(source_cats, fname)
             dst = os.path.join(validation_cats_dir, fname)
             shutil.copyfile(src, dst)
```

```
        fnames = ['{}.jpg'.format(i) for i in range(1500, 2000)]
        for fname in fnames:
            src = os.path.join(source_cats, fname)
            dst = os.path.join(test_cats_dir, fname)
            shutil.copyfile(src, dst)

        fnames = ['{}.jpg'.format(i) for i in range(1000)]
        for fname in fnames:
            src = os.path.join(source_dogs, fname)
            dst = os.path.join(train_dogs_dir, fname)
            shutil.copyfile(src, dst)

        fnames = ['{}.jpg'.format(i) for i in range(1000, 1500)]
        for fname in fnames:
            src = os.path.join(source_dogs, fname)
            dst = os.path.join(validation_dogs_dir, fname)
            shutil.copyfile(src, dst)

        fnames = ['{}.jpg'.format(i) for i in range(1500, 2000)]
        for fname in fnames:
            src = os.path.join(source_dogs, fname)
            dst = os.path.join(test_dogs_dir, fname)
            shutil.copyfile(src, dst)
```

```
In [13]: print('total training cat images: ', len(os.listdir(train_cats_dir)))
         print('total validation cat images: ', len(os.listdir(validation_cats_dir)))
         print('total test cat images: ', len(os.listdir(test_cats_dir)))

         print('total training dog images: ', len(os.listdir(train_dogs_dir)))
         print('total validation dog images: ', len(os.listdir(validation_dogs_dir)))
         print('total test dog images: ', len(os.listdir(test_dogs_dir)))
```

```
total training cat images:  1000
total validation cat images:  500
total test cat images:  500
total training dog images:  1000
total validation dog images:  500
total test dog images:  500
```

```
In [14]: from keras import layers
         from keras import models
```

```
Using TensorFlow backend.
```

```
In [15]: model = models.Sequential()
         model.add(layers.Conv2D(32, (3,3), activation='relu', input_shape=(150,150,3)))
         model.add(layers.MaxPooling2D((2,2)))
         model.add(layers.Conv2D(64, (3,3), activation='relu'))
         model.add(layers.MaxPooling2D((2,2)))
         model.add(layers.Conv2D(128, (3,3), activation='relu'))
         model.add(layers.MaxPooling2D((2,2)))
         model.add(layers.Conv2D(128, (3,3), activation='relu'))
         model.add(layers.MaxPooling2D((2,2)))
         model.add(layers.Flatten())
         model.add(layers.Dense(512, activation='relu'))
         model.add(layers.Dense(1, activation='sigmoid'))
```

```
In [16]: model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 148, 148, 32)      896
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 74, 74, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 72, 72, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 36, 36, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 34, 34, 128)       73856
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 17, 17, 128)       0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 15, 15, 128)       147584
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 7, 7, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 6272)              0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               3211776
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 513
=================================================================
Total params: 3,453,121
Trainable params: 3,453,121
Non-trainable params: 0
_________________________________________________________________
```

```
In [17]: from keras import optimizers

         model.compile(loss='binary_crossentropy',
                       optimizer=optimizers.RMSprop(lr=1e-4),
                       metrics=['acc'])
```

```
In [18]: from keras.preprocessing.image import ImageDataGenerator

         train_datagen = ImageDataGenerator(rescale=1./255)
         test_datagen = ImageDataGenerator(rescale=1./255)

         train_generator = train_datagen.flow_from_directory(
                 train_dir,
                 target_size=(150,150),
                 batch_size=20,
                 class_mode='binary')

         validation_generator = test_datagen.flow_from_directory(
                 validation_dir,
                 target_size=(150,150),
                 batch_size=20,
                 class_mode='binary')
```

```
Found 2000 images belonging to 2 classes.
Found 1000 images belonging to 2 classes.
```

```
In [19]: for data_batch, labels_batch in train_generator:
             print('data batch shape: ', data_batch.shape)
             print('labels batch shape', labels_batch.shape)
             break
```

```
data batch shape:  (20, 150, 150, 3)
labels batch shape (20,)
```

```
In [20]: history = model.fit_generator(
             train_generator,
             steps_per_epoch=100,
             epochs=30,
             validation_data=validation_generator,
             validation_steps=50)
```

```
Epoch 1/30
100/100 [==============================] - 13s 129ms/step - loss: 0.6931 - acc: 0.5305 - val_lo
Epoch 2/30
100/100 [==============================] - 12s 117ms/step - loss: 0.6649 - acc: 0.6085 - val_lo
Epoch 3/30
100/100 [==============================] - 12s 116ms/step - loss: 0.6116 - acc: 0.6685 - val_lo
Epoch 4/30
100/100 [==============================] - 12s 116ms/step - loss: 0.5700 - acc: 0.7065 - val_lo
Epoch 5/30
100/100 [==============================] - 12s 116ms/step - loss: 0.5367 - acc: 0.7280 - val_lo
Epoch 6/30
100/100 [==============================] - 12s 116ms/step - loss: 0.5041 - acc: 0.7550 - val_lo
```

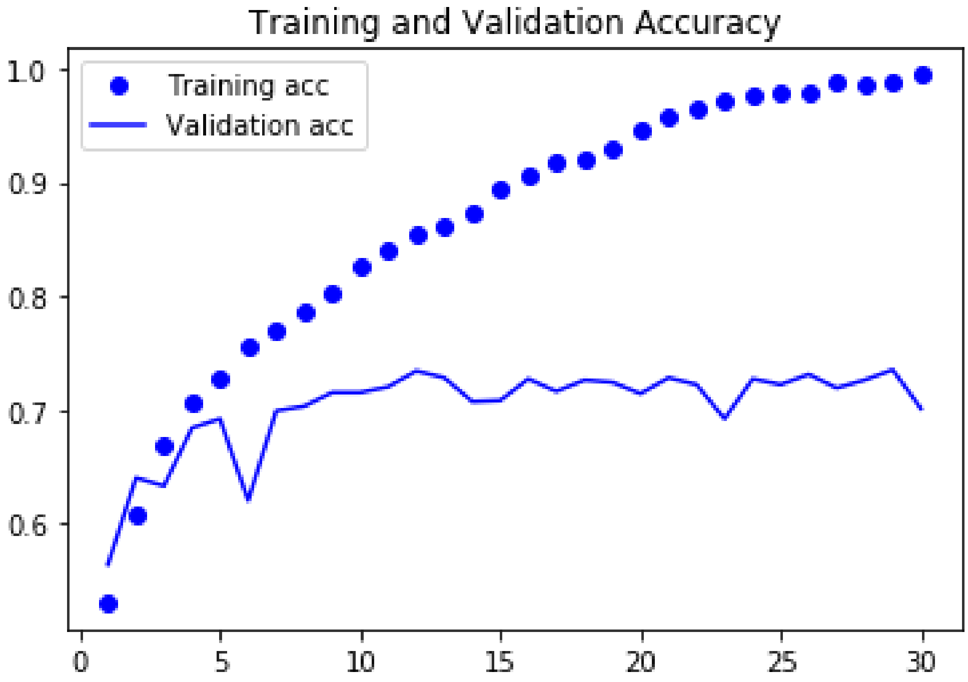
```
Epoch 7/30
100/100 [==============================] - 12s 116ms/step - loss: 0.4789 - acc: 0.7690 - val_lo
Epoch 8/30
100/100 [==============================] - 12s 117ms/step - loss: 0.4456 - acc: 0.7855 - val_lo
Epoch 9/30
100/100 [==============================] - 12s 116ms/step - loss: 0.4230 - acc: 0.8030 - val_lo
Epoch 10/30
100/100 [==============================] - 12s 115ms/step - loss: 0.3959 - acc: 0.8260 - val_lo
Epoch 11/30
100/100 [==============================] - 11s 115ms/step - loss: 0.3673 - acc: 0.8395 - val_lo
Epoch 12/30
100/100 [==============================] - 12s 116ms/step - loss: 0.3397 - acc: 0.8550 - val_lo
Epoch 13/30
100/100 [==============================] - 12s 116ms/step - loss: 0.3151 - acc: 0.8610 - val_lo
Epoch 14/30
100/100 [==============================] - 12s 116ms/step - loss: 0.2871 - acc: 0.8735 - val_lo
Epoch 15/30
100/100 [==============================] - 12s 116ms/step - loss: 0.2680 - acc: 0.8940 - val_lo
Epoch 16/30
100/100 [==============================] - 12s 115ms/step - loss: 0.2379 - acc: 0.9065 - val_lo
Epoch 17/30
100/100 [==============================] - 12s 117ms/step - loss: 0.2221 - acc: 0.9165 - val_lo
Epoch 18/30
100/100 [==============================] - 12s 116ms/step - loss: 0.1993 - acc: 0.9190 - val_lo
Epoch 19/30
100/100 [==============================] - 12s 116ms/step - loss: 0.1783 - acc: 0.9290 - val_lo
Epoch 20/30
100/100 [==============================] - 12s 115ms/step - loss: 0.1606 - acc: 0.9460 - val_lo
Epoch 21/30
100/100 [==============================] - 12s 116ms/step - loss: 0.1333 - acc: 0.9575 - val_lo
Epoch 22/30
100/100 [==============================] - 12s 116ms/step - loss: 0.1158 - acc: 0.9645 - val_lo
Epoch 23/30
100/100 [==============================] - 12s 116ms/step - loss: 0.0997 - acc: 0.9715 - val_lo
Epoch 24/30
100/100 [==============================] - 12s 116ms/step - loss: 0.0854 - acc: 0.9755 - val_lo
Epoch 25/30
100/100 [==============================] - 12s 117ms/step - loss: 0.0777 - acc: 0.9775 - val_lo
Epoch 26/30
100/100 [==============================] - 12s 116ms/step - loss: 0.0657 - acc: 0.9775 - val_lo
Epoch 27/30
100/100 [==============================] - 12s 116ms/step - loss: 0.0507 - acc: 0.9880 - val_lo
Epoch 28/30
100/100 [==============================] - 12s 116ms/step - loss: 0.0490 - acc: 0.9860 - val_lo
Epoch 29/30
100/100 [==============================] - 12s 116ms/step - loss: 0.0403 - acc: 0.9880 - val_lo
Epoch 30/30
100/100 [==============================] - 12s 116ms/step - loss: 0.0289 - acc: 0.9945 - val_lo
```

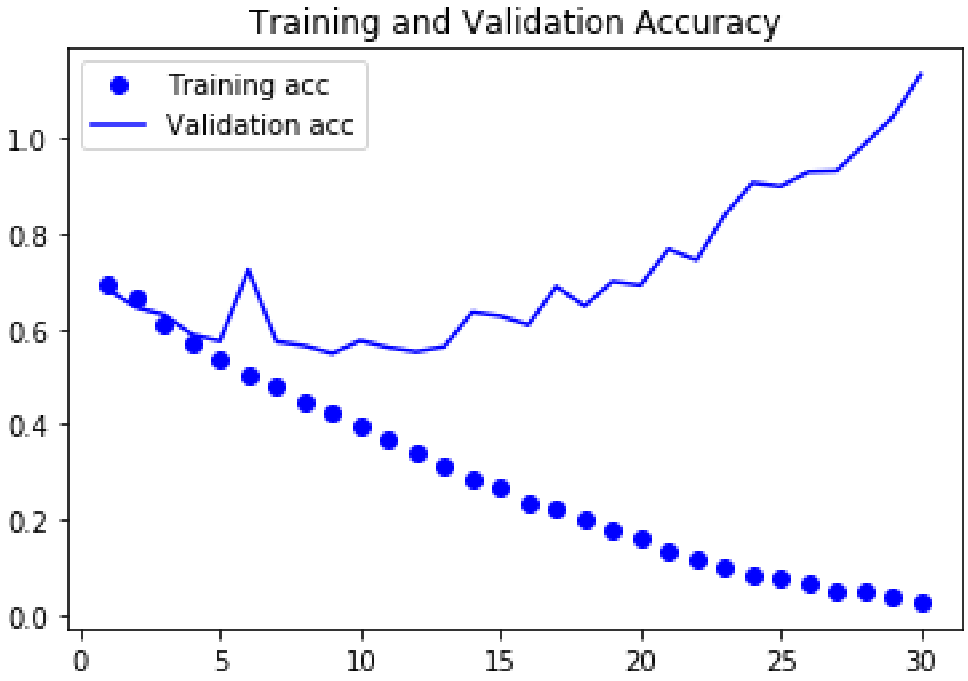
```
In [21]: model.save('cats_and_dogs_small.h5')
```

```
In [22]: # lets see if we can get some data feedback
         model_keys = history.history.keys()
         print(model_keys)
```

```
dict_keys(['acc', 'val_acc', 'loss', 'val_loss'])
```

```
In [23]: acc = history.history['acc']
         val_acc = history.history['val_acc']
         loss = history.history['loss']
         val_loss = history.history['val_loss']

         epochs = range(1, len(acc) + 1)

         plt.plot(epochs, acc, 'bo', label='Training acc')
         plt.plot(epochs, val_acc, 'b', label='Validation acc')
         plt.title("Training and Validation Accuracy")

         plt.legend()
         plt.figure()


         plt.plot(epochs, loss, 'bo', label='Training acc')
         plt.plot(epochs, val_loss, 'b', label='Validation acc')
         plt.title("Training and Validation Accuracy")

         plt.legend()
         plt.figure()

         plt.show()
```


Training and Validation Accuracy
● Training acc
Validation acc
1.0
0.9
0.8
0.7
0.6
0
5
10
15
20
25
30


Training and Validation Accuracy
● Training acc
Validation acc
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30

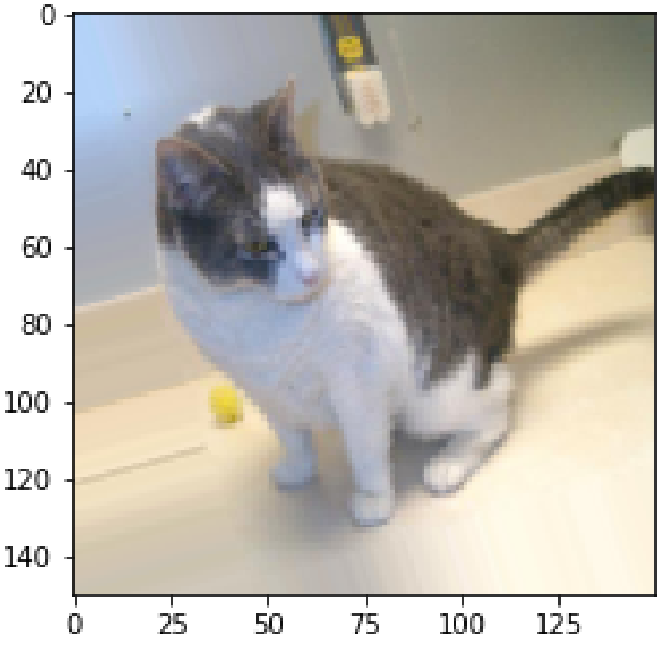
```
<Figure size 432x288 with 0 Axes>
```

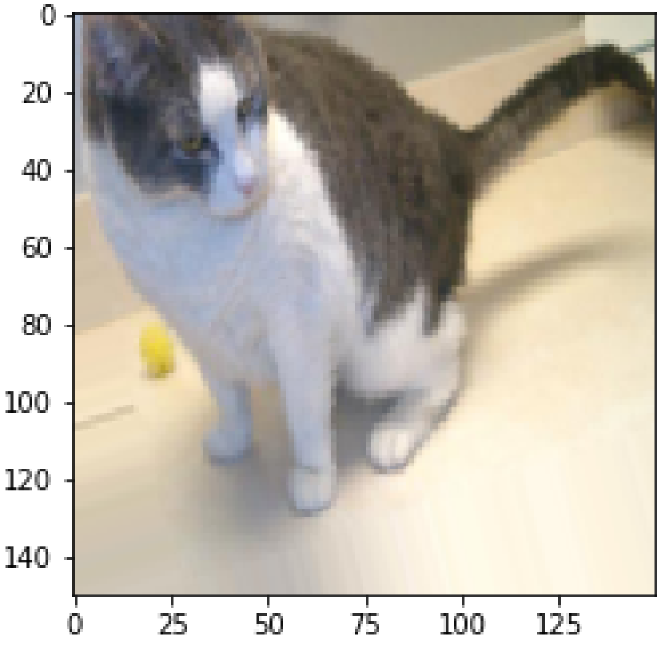
In [24]:

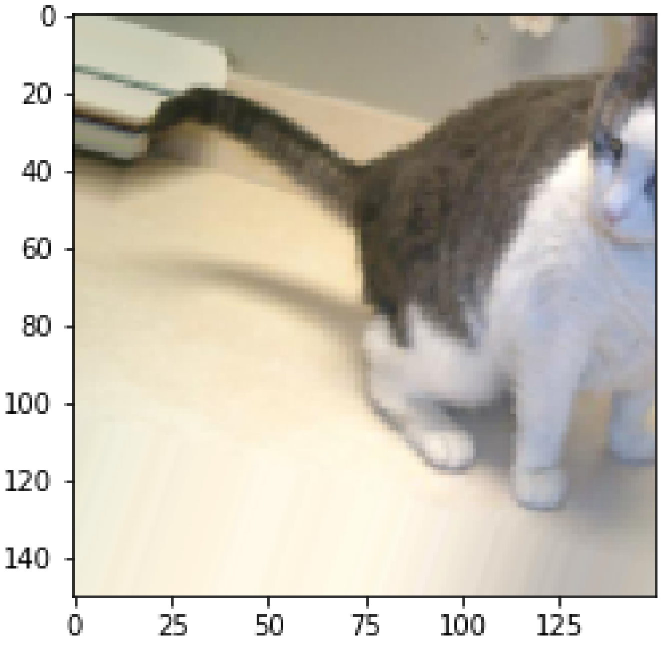
```
# These plots are very characteristic of overfitting
# Let's fix this using data augmentation
# this is just an example we wont actually be using datagen

datagen = ImageDataGenerator(
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')
```

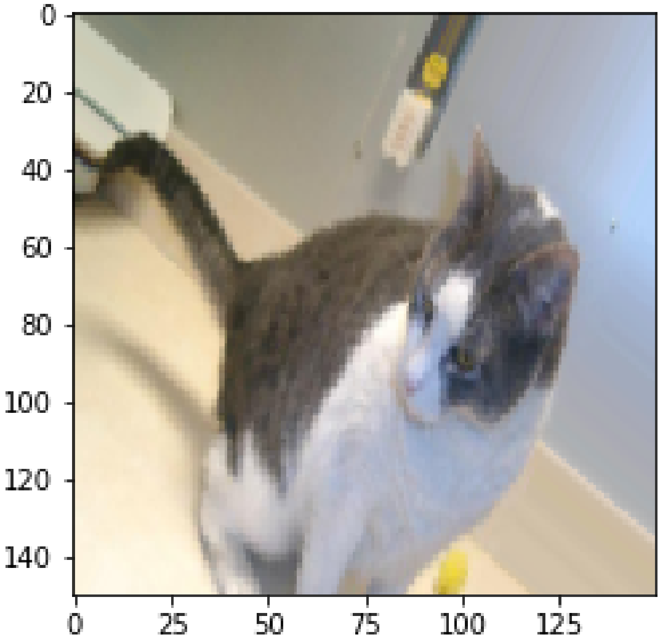
In [25]:

```
from keras.preprocessing import image

fnames = [os.path.join(train_cats_dir, fname) for
         fname in os.listdir(train_cats_dir)]

img_path = fnames[3]

img = image.load_img(img_path, target_size=(150, 150))

x = image.img_to_array(img)

x = x.reshape((1,) + x.shape)

i = 0
for batch in datagen.flow(x, batch_size=1):
    plt.figure(i)
    imgplot = plt.imshow(image.array_to_img(batch[0]))
    i += 1
    if i % 4 == 0:
        break
plt.show()
```


0
20
40
60
80
100
120
140
0
25
50
75
100
125


0
20
40
60
80
100
120
140
0
25
50
75
100
125





```
In [27]: # We will also be using dropout to further increase generalization

         model = models.Sequential()
         model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)))
         model.add(layers.MaxPooling2D((2, 2)))
         model.add(layers.Conv2D(64, (3, 3), activation='relu'))
         model.add(layers.MaxPooling2D((2, 2)))
         model.add(layers.Conv2D(128, (3, 3), activation='relu'))
         model.add(layers.MaxPooling2D((2, 2)))
         model.add(layers.Conv2D(128, (3, 3), activation='relu'))
         model.add(layers.MaxPooling2D((2, 2)))
         model.add(layers.Flatten())
         model.add(layers.Dropout(0.5))
         model.add(layers.Dense(512, activation='relu'))
         model.add(layers.Dense(1, activation='sigmoid'))

         model.compile(loss='binary_crossentropy',
                       optimizer=optimizers.RMSprop(lr=1e-4),
                       metrics=['acc'])
```

```
In [28]: model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_9 (Conv2D)            (None, 148, 148, 32)      896
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 74, 74, 32)        0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 72, 72, 64)        18496
_________________________________________________________________
max_pooling2d_10 (MaxPooling (None, 36, 36, 64)        0
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 34, 34, 128)       73856
_________________________________________________________________
max_pooling2d_11 (MaxPooling (None, 17, 17, 128)       0
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 15, 15, 128)       147584
_________________________________________________________________
max_pooling2d_12 (MaxPooling (None, 7, 7, 128)         0
_________________________________________________________________
flatten_3 (Flatten)          (None, 6272)              0
_________________________________________________________________
dropout_2 (Dropout)          (None, 6272)              0
_________________________________________________________________
dense_4 (Dense)              (None, 512)               3211776
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 513
=================================================================
Total params: 3,453,121
```

```
Trainable params: 3,453,121
Non-trainable params: 0
_________________________________________________________________
```

```
In [35]: # our new generator with data augmentation
         train_datagen = ImageDataGenerator(
             rescale=1./255,
             rotation_range=40,
             width_shift_range=0.2,
             height_shift_range=0.2,
             shear_range=0.2,
             zoom_range=0.2,
             horizontal_flip=True)

         test_datagen = ImageDataGenerator(
             rescale=1./255)                                           # notice we won't be au

         train_generator = train_datagen.flow_from_directory(
             train_dir,
             target_size=(150, 150),
             batch_size=32,
             class_mode='binary')

         validation_generator = test_datagen.flow_from_directory(
             validation_dir,
             target_size=(150, 150),
             batch_size=32,
             class_mode='binary')
```

```
Found 2000 images belonging to 2 classes.
Found 1000 images belonging to 2 classes.
```

```
In [36]: history = model.fit_generator(
             train_generator,
             steps_per_epoch=100,
             epochs=100,
             validation_data=validation_generator,
             validation_steps=50)
```

```
Epoch 1/100
100/100 [==============================] - 29s 289ms/step - loss: 0.6915 - acc: 0.5187 - val_lo
Epoch 2/100
100/100 [==============================] - 26s 264ms/step - loss: 0.6762 - acc: 0.5650 - val_lo
Epoch 3/100
100/100 [==============================] - 26s 263ms/step - loss: 0.6633 - acc: 0.5859 - val_lo
Epoch 4/100
100/100 [==============================] - 27s 270ms/step - loss: 0.6516 - acc: 0.5950 - val_lo
```

```
Epoch 5/100
100/100 [==============================] - 26s 265ms/step - loss: 0.6376 - acc: 0.6119 - val_lo
Epoch 6/100
100/100 [==============================] - 29s 291ms/step - loss: 0.6150 - acc: 0.6628 - val_lo
Epoch 7/100
100/100 [==============================] - 27s 267ms/step - loss: 0.6089 - acc: 0.6588 - val_lo
Epoch 8/100
100/100 [==============================] - 28s 277ms/step - loss: 0.5889 - acc: 0.6803 - val_lo
Epoch 9/100
100/100 [==============================] - 26s 264ms/step - loss: 0.5896 - acc: 0.6816 - val_lo
Epoch 10/100
100/100 [==============================] - 26s 264ms/step - loss: 0.5804 - acc: 0.6916 - val_lo
Epoch 11/100
100/100 [==============================] - 28s 275ms/step - loss: 0.5754 - acc: 0.6938 - val_lo
Epoch 12/100
100/100 [==============================] - 27s 266ms/step - loss: 0.5639 - acc: 0.6975 - val_lo
Epoch 13/100
100/100 [==============================] - 27s 269ms/step - loss: 0.5564 - acc: 0.7022 - val_lo
Epoch 14/100
100/100 [==============================] - 27s 266ms/step - loss: 0.5585 - acc: 0.7087 - val_lo
Epoch 15/100
100/100 [==============================] - 27s 275ms/step - loss: 0.5575 - acc: 0.7175 - val_lo
Epoch 16/100
100/100 [==============================] - 27s 268ms/step - loss: 0.5516 - acc: 0.7225 - val_lo
Epoch 17/100
100/100 [==============================] - 26s 265ms/step - loss: 0.5367 - acc: 0.7281 - val_lo
Epoch 18/100
100/100 [==============================] - 29s 288ms/step - loss: 0.5305 - acc: 0.7310 - val_lo
Epoch 19/100
100/100 [==============================] - 27s 265ms/step - loss: 0.5392 - acc: 0.7284 - val_lo
Epoch 20/100
100/100 [==============================] - 27s 268ms/step - loss: 0.5253 - acc: 0.7344 - val_lo
Epoch 21/100
100/100 [==============================] - 27s 266ms/step - loss: 0.5269 - acc: 0.7469 - val_lo
Epoch 22/100
100/100 [==============================] - 27s 272ms/step - loss: 0.5271 - acc: 0.7350 - val_lo
Epoch 23/100
100/100 [==============================] - 28s 281ms/step - loss: 0.5212 - acc: 0.7384 - val_lo
Epoch 24/100
100/100 [==============================] - 28s 278ms/step - loss: 0.5140 - acc: 0.7453 - val_lo
Epoch 25/100
100/100 [==============================] - 26s 265ms/step - loss: 0.5112 - acc: 0.7462 - val_lo
Epoch 26/100
100/100 [==============================] - 26s 265ms/step - loss: 0.5133 - acc: 0.7416 - val_lo
Epoch 27/100
100/100 [==============================] - 27s 265ms/step - loss: 0.4994 - acc: 0.7591 - val_lo
Epoch 28/100
100/100 [==============================] - 27s 275ms/step - loss: 0.5010 - acc: 0.7559 - val_lo
```

```
Epoch 29/100
100/100 [==============================] - 27s 271ms/step - loss: 0.4908 - acc: 0.7553 - val_l
Epoch 30/100
100/100 [==============================] - 26s 263ms/step - loss: 0.4943 - acc: 0.7562 - val_l
Epoch 31/100
100/100 [==============================] - 28s 276ms/step - loss: 0.4887 - acc: 0.7559 - val_l
Epoch 32/100
100/100 [==============================] - 26s 265ms/step - loss: 0.4879 - acc: 0.7678 - val_l
Epoch 33/100
100/100 [==============================] - 26s 264ms/step - loss: 0.4828 - acc: 0.7600 - val_l
Epoch 34/100
100/100 [==============================] - 26s 264ms/step - loss: 0.4706 - acc: 0.7759 - val_l
Epoch 35/100
100/100 [==============================] - 29s 285ms/step - loss: 0.4776 - acc: 0.7706 - val_l
Epoch 36/100
100/100 [==============================] - 27s 269ms/step - loss: 0.4694 - acc: 0.7756 - val_l
Epoch 37/100
100/100 [==============================] - 27s 265ms/step - loss: 0.4773 - acc: 0.7609 - val_l
Epoch 38/100
100/100 [==============================] - 27s 274ms/step - loss: 0.4654 - acc: 0.7788 - val_l
Epoch 39/100
100/100 [==============================] - 26s 265ms/step - loss: 0.4660 - acc: 0.7722 - val_l
Epoch 40/100
100/100 [==============================] - 29s 292ms/step - loss: 0.4652 - acc: 0.7803 - val_l
Epoch 41/100
100/100 [==============================] - 26s 265ms/step - loss: 0.4576 - acc: 0.7859 - val_l
Epoch 42/100
100/100 [==============================] - 26s 265ms/step - loss: 0.4682 - acc: 0.7809 - val_l
Epoch 43/100
100/100 [==============================] - 27s 266ms/step - loss: 0.4495 - acc: 0.7816 - val_l
Epoch 44/100
100/100 [==============================] - 26s 265ms/step - loss: 0.4650 - acc: 0.7772 - val_l
Epoch 45/100
100/100 [==============================] - 28s 280ms/step - loss: 0.4433 - acc: 0.7813 - val_l
Epoch 46/100
100/100 [==============================] - 27s 265ms/step - loss: 0.4465 - acc: 0.7912 - val_l
Epoch 47/100
100/100 [==============================] - 27s 274ms/step - loss: 0.4473 - acc: 0.7891 - val_l
Epoch 48/100
100/100 [==============================] - 26s 263ms/step - loss: 0.4425 - acc: 0.7913 - val_l
Epoch 49/100
100/100 [==============================] - 27s 267ms/step - loss: 0.4484 - acc: 0.7934 - val_l
Epoch 50/100
100/100 [==============================] - 26s 265ms/step - loss: 0.4186 - acc: 0.8112 - val_l
Epoch 51/100
100/100 [==============================] - 26s 264ms/step - loss: 0.4390 - acc: 0.7978 - val_l
Epoch 52/100
100/100 [==============================] - 29s 286ms/step - loss: 0.4378 - acc: 0.7891 - val_l
```

```
Epoch 53/100
100/100 [==============================] - 27s 267ms/step - loss: 0.4241 - acc: 0.8028 - val_lo
Epoch 54/100
100/100 [==============================] - 27s 271ms/step - loss: 0.4377 - acc: 0.7953 - val_lo
Epoch 55/100
100/100 [==============================] - 26s 265ms/step - loss: 0.4239 - acc: 0.8031 - val_lo
Epoch 56/100
100/100 [==============================] - 28s 276ms/step - loss: 0.4272 - acc: 0.7935 - val_lo
Epoch 57/100
100/100 [==============================] - 28s 277ms/step - loss: 0.4352 - acc: 0.8019 - val_lo
Epoch 58/100
100/100 [==============================] - 27s 266ms/step - loss: 0.4168 - acc: 0.8075 - val_lo
Epoch 59/100
100/100 [==============================] - 26s 265ms/step - loss: 0.4272 - acc: 0.8062 - val_lo
Epoch 60/100
100/100 [==============================] - 27s 266ms/step - loss: 0.4059 - acc: 0.8147 - val_lo
Epoch 61/100
100/100 [==============================] - 27s 271ms/step - loss: 0.4112 - acc: 0.8059 - val_lo
Epoch 62/100
100/100 [==============================] - 27s 271ms/step - loss: 0.4150 - acc: 0.8047 - val_lo
Epoch 63/100
100/100 [==============================] - 28s 275ms/step - loss: 0.4159 - acc: 0.8031 - val_lo
Epoch 64/100
100/100 [==============================] - 26s 264ms/step - loss: 0.3974 - acc: 0.8216 - val_lo
Epoch 65/100
100/100 [==============================] - 26s 263ms/step - loss: 0.4028 - acc: 0.8087 - val_lo
Epoch 66/100
100/100 [==============================] - 27s 265ms/step - loss: 0.4010 - acc: 0.8163 - val_lo
Epoch 67/100
100/100 [==============================] - 26s 264ms/step - loss: 0.3933 - acc: 0.8241 - val_lo
Epoch 68/100
100/100 [==============================] - 27s 268ms/step - loss: 0.3941 - acc: 0.8184 - val_lo
Epoch 69/100
100/100 [==============================] - 28s 283ms/step - loss: 0.3897 - acc: 0.8237 - val_lo
Epoch 70/100
100/100 [==============================] - 27s 274ms/step - loss: 0.3856 - acc: 0.8197 - val_lo
Epoch 71/100
100/100 [==============================] - 26s 263ms/step - loss: 0.3861 - acc: 0.8312 - val_lo
Epoch 72/100
100/100 [==============================] - 28s 280ms/step - loss: 0.3909 - acc: 0.8281 - val_lo
Epoch 73/100
100/100 [==============================] - 26s 264ms/step - loss: 0.3918 - acc: 0.8219 - val_lo
Epoch 74/100
100/100 [==============================] - 27s 275ms/step - loss: 0.3946 - acc: 0.8209 - val_lo
Epoch 75/100
100/100 [==============================] - 27s 267ms/step - loss: 0.3924 - acc: 0.8175 - val_lo
Epoch 76/100
100/100 [==============================] - 26s 265ms/step - loss: 0.3791 - acc: 0.8263 - val_lo
```

```
Epoch 77/100
100/100 [==============================] - 27s 272ms/step - loss: 0.3796 - acc: 0.8272 - val_lo
Epoch 78/100
100/100 [==============================] - 26s 264ms/step - loss: 0.3831 - acc: 0.8256 - val_lo
Epoch 79/100
100/100 [==============================] - 28s 279ms/step - loss: 0.3649 - acc: 0.8359 - val_lo
Epoch 80/100
100/100 [==============================] - 26s 263ms/step - loss: 0.3854 - acc: 0.8241 - val_lo
Epoch 81/100
100/100 [==============================] - 29s 287ms/step - loss: 0.3622 - acc: 0.8344 - val_lo
Epoch 82/100
100/100 [==============================] - 26s 263ms/step - loss: 0.3815 - acc: 0.8247 - val_lo
Epoch 83/100
100/100 [==============================] - 26s 265ms/step - loss: 0.3755 - acc: 0.8297 - val_lo
Epoch 84/100
100/100 [==============================] - 27s 268ms/step - loss: 0.3483 - acc: 0.8372 - val_lo
Epoch 85/100
100/100 [==============================] - 26s 265ms/step - loss: 0.3740 - acc: 0.8331 - val_lo
Epoch 86/100
100/100 [==============================] - 29s 290ms/step - loss: 0.3612 - acc: 0.8335 - val_lo
Epoch 87/100
100/100 [==============================] - 26s 264ms/step - loss: 0.3515 - acc: 0.8453 - val_lo
Epoch 88/100
100/100 [==============================] - 28s 277ms/step - loss: 0.3558 - acc: 0.8359 - val_lo
Epoch 89/100
100/100 [==============================] - 26s 264ms/step - loss: 0.3468 - acc: 0.8503 - val_lo
Epoch 90/100
100/100 [==============================] - 26s 264ms/step - loss: 0.3439 - acc: 0.8516 - val_lo
Epoch 91/100
100/100 [==============================] - 27s 274ms/step - loss: 0.3553 - acc: 0.8456 - val_lo
Epoch 92/100
100/100 [==============================] - 27s 266ms/step - loss: 0.3449 - acc: 0.8400 - val_lo
Epoch 93/100
100/100 [==============================] - 27s 269ms/step - loss: 0.3477 - acc: 0.8450 - val_lo
Epoch 94/100
100/100 [==============================] - 26s 265ms/step - loss: 0.3376 - acc: 0.8491 - val_lo
Epoch 95/100
100/100 [==============================] - 28s 278ms/step - loss: 0.3461 - acc: 0.8387 - val_lo
Epoch 96/100
100/100 [==============================] - 26s 265ms/step - loss: 0.3394 - acc: 0.8541 - val_lo
Epoch 97/100
100/100 [==============================] - 27s 266ms/step - loss: 0.3386 - acc: 0.8538 - val_lo
Epoch 98/100
100/100 [==============================] - 28s 283ms/step - loss: 0.3346 - acc: 0.8563 - val_lo
Epoch 99/100
100/100 [==============================] - 27s 265ms/step - loss: 0.3221 - acc: 0.8534 - val_lo
Epoch 100/100
100/100 [==============================] - 27s 267ms/step - loss: 0.3315 - acc: 0.8531 - val_lo
```

```
In [37]: model.save('cats_and_dogs_small_2.h5')
```