cat-vs-dog

June 9, 2020

[137]:

```
import os,shutil
```

[138]:

```
original_dataset_dir = 'D:\Academics\Deep learning Adnan\datasets\cat-vs-dog'
```

[139]:

```
base_dir = 'D:\Academics\Deep learning Adnan\datasets\cats_and_dogs_small'
```

[141]:

```
os.mkdir(base_dir)
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)
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)
fnames = ['cat.{}.jpg'.format(i) for i in range(1000)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(train_cats_dir, fname)
    shutil.copyfile(src, dst)
fnames = ['cat.{}.jpg'.format(i) for i in range(1000, 1500)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(validation_cats_dir, fname)
    shutil.copyfile(src, dst)
fnames = ['cat.{}.jpg'.format(i) for i in range(1500, 2000)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(test_cats_dir, fname)
    shutil.copyfile(src, dst)
    fnames = ['dog.{}.jpg'.format(i) for i in range(1000)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(train_dogs_dir, fname)
    shutil.copyfile(src, dst)
fnames = ['dog.{}.jpg'.format(i) for i in range(1000, 1500)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(validation_dogs_dir, fname)
    shutil.copyfile(src, dst)
fnames = ['dog.{}.jpg'.format(i) for i in range(1500, 2000)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(test_dogs_dir, fname)
    shutil.copyfile(src, dst)
```

[142]:

```
print('total training cat images:', len(os.listdir(train_cats_dir)))
```

```
total training cat images: 1000
```

[143]:

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

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

[144]:

```
from keras import layers
from keras import models
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'))
```

```
Using TensorFlow backend.
```

[145]:

```
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
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
_________________________________________________________________
```

[146]:

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

[147]:

```
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.
```

[148]:

```
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,)
```

[149]:

```
history = model.fit_generator(
train_generator,
steps_per_epoch=100,
epochs=30,
validation_data=validation_generator,
validation_steps=50)
```

```
Epoch 1/30
100/100 [==============================] - 69s 692ms/step - loss: 0.6907 - acc:
0.5305 - val_loss: 0.6789 - val_acc: 0.5220
Epoch 2/30
100/100 [==============================] - 68s 676ms/step - loss: 0.6603 - acc:
0.6120 - val_loss: 0.6212 - val_acc: 0.6440
Epoch 3/30
100/100 [==============================] - 64s 643ms/step - loss: 0.6066 - acc:
0.6785 - val_loss: 0.5561 - val_acc: 0.6810
Epoch 4/30
100/100 [==============================] - 66s 665ms/step - loss: 0.5646 - acc:
0.7125 - val_loss: 0.5285 - val_acc: 0.6240
Epoch 5/30
100/100 [==============================] - 70s 701ms/step - loss: 0.5358 - acc:
0.7310 - val_loss: 0.4909 - val_acc: 0.6920
Epoch 6/30
100/100 [==============================] - 71s 713ms/step - loss: 0.5111 - acc:
0.7410 - val_loss: 0.5621 - val_acc: 0.7050
Epoch 7/30
100/100 [==============================] - 69s 693ms/step - loss: 0.4780 - acc:
```

```
0.7695 - val_loss: 0.5460 - val_acc: 0.6910
Epoch 8/30
100/100 [==============================] - 69s 688ms/step - loss: 0.4586 - acc:
0.7765 - val_loss: 0.5783 - val_acc: 0.7140
Epoch 9/30
100/100 [==============================] - 69s 686ms/step - loss: 0.4305 - acc:
0.7925 - val_loss: 0.4107 - val_acc: 0.7080
Epoch 10/30
100/100 [==============================] - 70s 703ms/step - loss: 0.4009 - acc:
0.8155 - val_loss: 0.5871 - val_acc: 0.7110
Epoch 11/30
100/100 [==============================] - 70s 701ms/step - loss: 0.3834 - acc:
0.8300 - val_loss: 0.4446 - val_acc: 0.7230
Epoch 12/30
100/100 [==============================] - 70s 699ms/step - loss: 0.3588 - acc:
0.8415 - val_loss: 0.7586 - val_acc: 0.7300
Epoch 13/30
100/100 [==============================] - 70s 703ms/step - loss: 0.3374 - acc:
0.8495 - val_loss: 0.6121 - val_acc: 0.6990
Epoch 14/30
100/100 [==============================] - 71s 707ms/step - loss: 0.3083 - acc:
0.8655 - val_loss: 0.4961 - val_acc: 0.7320
Epoch 15/30
100/100 [==============================] - 70s 701ms/step - loss: 0.2873 - acc:
0.8815 - val_loss: 0.7542 - val_acc: 0.7200
Epoch 16/30
100/100 [==============================] - 70s 705ms/step - loss: 0.2661 - acc:
0.8905 - val_loss: 0.5312 - val_acc: 0.6900
Epoch 17/30
100/100 [==============================] - 70s 701ms/step - loss: 0.2458 - acc:
0.9050 - val_loss: 0.3891 - val_acc: 0.7170
Epoch 18/30
100/100 [==============================] - 70s 703ms/step - loss: 0.2287 - acc:
0.9120 - val_loss: 0.9130 - val_acc: 0.7280
Epoch 19/30
100/100 [==============================] - 71s 705ms/step - loss: 0.1996 - acc:
0.9225 - val_loss: 0.4524 - val_acc: 0.7400
Epoch 20/30
100/100 [==============================] - 70s 698ms/step - loss: 0.1795 - acc:
0.9355 - val_loss: 0.7049 - val_acc: 0.7340
Epoch 21/30
100/100 [==============================] - 69s 689ms/step - loss: 0.1571 - acc:
0.9490 - val_loss: 0.8933 - val_acc: 0.7330
Epoch 22/30
100/100 [==============================] - 69s 691ms/step - loss: 0.1380 - acc:
0.9515 - val_loss: 1.0607 - val_acc: 0.7140
Epoch 23/30
100/100 [==============================] - 71s 707ms/step - loss: 0.1267 - acc:
```

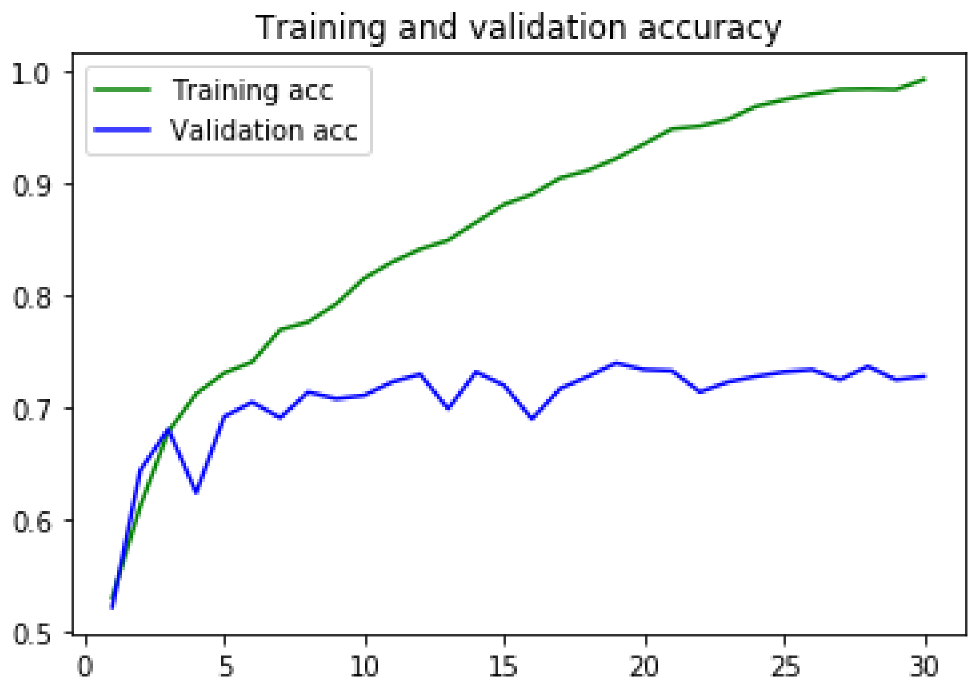
```
0.9575 - val_loss: 0.8147 - val_acc: 0.7230
Epoch 24/30
100/100 [==============================] - 68s 679ms/step - loss: 0.1049 - acc:
0.9690 - val_loss: 1.1030 - val_acc: 0.7280
Epoch 25/30
100/100 [==============================] - 70s 703ms/step - loss: 0.0910 - acc:
0.9750 - val_loss: 1.0858 - val_acc: 0.7320
Epoch 26/30
100/100 [==============================] - 70s 701ms/step - loss: 0.0794 - acc:
0.9800 - val_loss: 0.2945 - val_acc: 0.7340
Epoch 27/30
100/100 [==============================] - 70s 704ms/step - loss: 0.0666 - acc:
0.9840 - val_loss: 0.9596 - val_acc: 0.7250
Epoch 28/30
100/100 [==============================] - 70s 704ms/step - loss: 0.0610 - acc:
0.9845 - val_loss: 1.0316 - val_acc: 0.7370
Epoch 29/30
100/100 [==============================] - 70s 702ms/step - loss: 0.0520 - acc:
0.9840 - val_loss: 0.2829 - val_acc: 0.7250
Epoch 30/30
100/100 [==============================] - 70s 696ms/step - loss: 0.0385 - acc:
0.9930 - val_loss: 0.5773 - val_acc: 0.7280
```

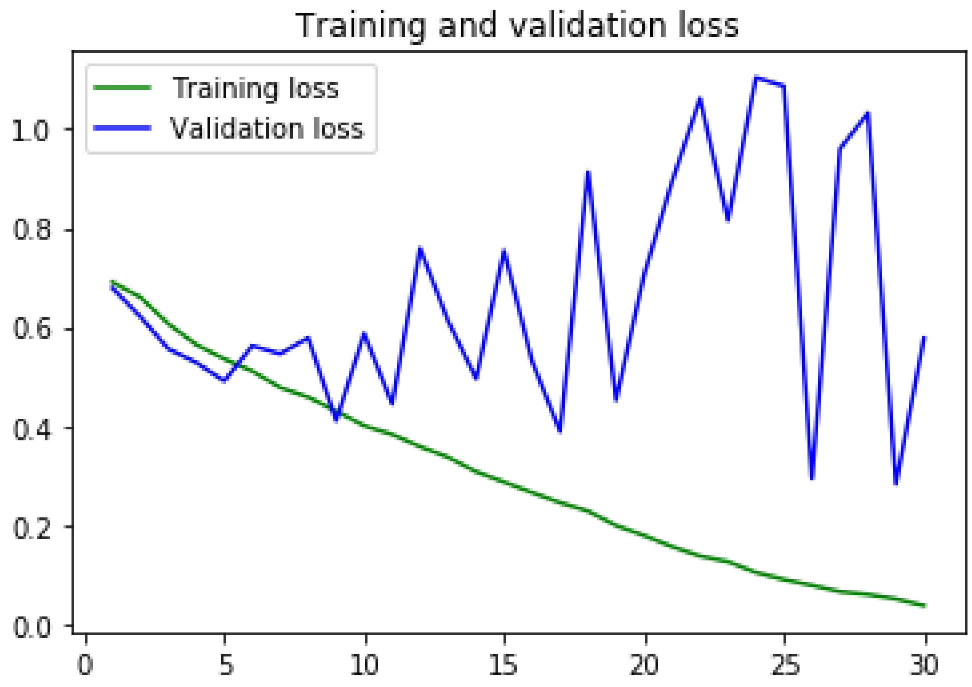
[150]:

```python
model.save('cats_and_dogs_small_1.h5')
```

[151]:

```python
import matplotlib.pyplot as plt
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, 'g', 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, 'g', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()
plt.show()
```


Training and validation accuracy
Training acc
Validation acc
1.0
0.9
0.8
0.7
0.6
0.5
0
5
10
15
20
25
30


Training and validation loss
Training loss
Validation loss
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30

```
[ ]:
```