classifier

January 23, 2020

[5]:
```
import numpy as np
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dropout, Flatten, Dense
from tensorflow.keras import applications
from tensorflow.keras.utils import to_categorical
from tensorflow.keras import optimizers
from tensorflow.keras import backend as K
from tensorflow.keras.utils import plot_model
import matplotlib.pyplot as plt
```

[6]:
```
img_width, img_height = 150, 150

train_data_dir = '/home/user/     /convnets/transfer-learning-keras/dataset/
 ↪training'
validation_data_dir = '/home/user/     /convnets/transfer-learning-keras/
 ↪dataset/validation'
evaluation_data_dir = '/home/user/     /convnets/transfer-learning-keras/
 ↪dataset/evaluation'
nb_train_samples = 3000
nb_validation_samples = 1000
nb_evaluation_samples = 1000
epochs = 20
batch_size = 20
```

[11]:
```
train_datagen = ImageDataGenerator(
    rescale=1. / 255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True)

# this is the augmentation configuration we will use for testing:
# only rescaling
test_datagen = ImageDataGenerator(rescale=1. / 255)

train_generator = train_datagen.flow_from_directory(
    train_data_dir,
```

```
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='binary')

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

if K.image_data_format() == 'channels_first':
    input_shape = (3, img_width, img_height)
else:
    input_shape = (img_width, img_height, 3)

model = Sequential([
    Flatten(),
    Dense(128, activation='relu'),
    Dense(1, activation='sigmoid')
])

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

history = model.fit_generator(
    train_generator,
    steps_per_epoch=nb_train_samples // batch_size,
    epochs=epochs,
    validation_data=validation_generator,
    validation_steps=nb_validation_samples // batch_size)
```

```
Found 3000 images belonging to 2 classes.
Found 1000 images belonging to 2 classes.
Epoch 1/20
149/150 [============================>.] - ETA: 0s - loss: 1.8926 - acc:
0.6457Epoch 1/20
150/150 [==============================] - 35s 234ms/step - loss: 1.8850 - acc:
0.6462 - val_loss: 1.4257 - val_acc: 0.6170
Epoch 2/20
149/150 [============================>.] - ETA: 0s - loss: 0.8491 - acc:
0.7020Epoch 1/20
150/150 [==============================] - 33s 220ms/step - loss: 0.8475 - acc:
0.7021 - val_loss: 0.9023 - val_acc: 0.6990
Epoch 3/20
149/150 [============================>.] - ETA: 0s - loss: 0.6925 - acc:
0.7169Epoch 1/20
```

```
150/150 [==============================] - 33s 223ms/step - loss: 0.6952 - acc:
0.7172 - val_loss: 0.9170 - val_acc: 0.6870
Epoch 4/20
149/150 [============================>.] - ETA: 0s - loss: 0.7860 - acc:
0.7000Epoch 1/20
150/150 [==============================] - 33s 222ms/step - loss: 0.7830 - acc:
0.7004 - val_loss: 0.8221 - val_acc: 0.7120
Epoch 5/20
149/150 [============================>.] - ETA: 0s - loss: 0.9393 - acc:
0.7266Epoch 1/20
150/150 [==============================] - 33s 222ms/step - loss: 0.9360 - acc:
0.7264 - val_loss: 1.1079 - val_acc: 0.6940
Epoch 6/20
149/150 [============================>.] - ETA: 0s - loss: 0.8922 - acc:
0.6992Epoch 1/20
150/150 [==============================] - 32s 216ms/step - loss: 0.8897 - acc:
0.6996 - val_loss: 0.6244 - val_acc: 0.7670
Epoch 7/20
149/150 [============================>.] - ETA: 0s - loss: 0.7999 - acc:
0.7135Epoch 1/20
150/150 [==============================] - 33s 221ms/step - loss: 0.8003 - acc:
0.7135 - val_loss: 0.8134 - val_acc: 0.7120
Epoch 8/20
149/150 [============================>.] - ETA: 0s - loss: 0.5672 - acc:
0.7765Epoch 1/20
150/150 [==============================] - 32s 214ms/step - loss: 0.5678 - acc:
0.7763 - val_loss: 0.7777 - val_acc: 0.6770
Epoch 9/20
149/150 [============================>.] - ETA: 0s - loss: 0.6717 - acc:
0.7389Epoch 1/20
150/150 [==============================] - 35s 233ms/step - loss: 0.6726 - acc:
0.7389 - val_loss: 0.7553 - val_acc: 0.7100
Epoch 10/20
149/150 [============================>.] - ETA: 0s - loss: 0.7376 - acc:
0.7182Epoch 1/20
150/150 [==============================] - 33s 217ms/step - loss: 0.7344 - acc:
0.7184 - val_loss: 0.5775 - val_acc: 0.7710
Epoch 11/20
149/150 [============================>.] - ETA: 0s - loss: 0.5760 - acc:
0.7793Epoch 1/20
150/150 [==============================] - 33s 223ms/step - loss: 0.5755 - acc:
0.7790 - val_loss: 0.6970 - val_acc: 0.7320
Epoch 12/20
149/150 [============================>.] - ETA: 0s - loss: 0.6098 - acc: 0.7504-
EEpoch 1/20
150/150 [==============================] - 32s 216ms/step - loss: 0.6083 - acc:
0.7505 - val_loss: 0.6629 - val_acc: 0.7150
Epoch 13/20
```

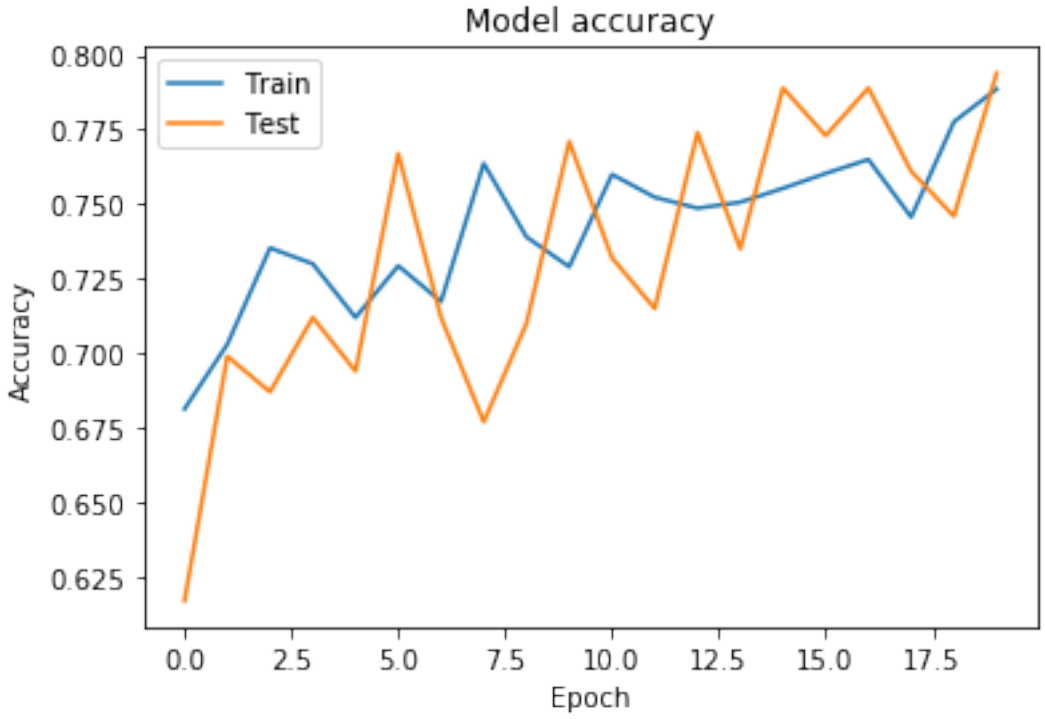
```
149/150 [============================>.] - ETA: 0s - loss: 0.5906 - acc:
0.7584Epoch 1/20
150/150 [==============================] - 33s 222ms/step - loss: 0.5905 - acc:
0.7583 - val_loss: 0.5679 - val_acc: 0.7740
Epoch 14/20
149/150 [============================>.] - ETA: 0s - loss: 0.5657 - acc:
0.7699Epoch 1/20
150/150 [==============================] - 33s 220ms/step - loss: 0.5676 - acc:
0.7696 - val_loss: 0.5828 - val_acc: 0.7350
Epoch 15/20
149/150 [============================>.] - ETA: 0s - loss: 0.5593 - acc:
0.7453Epoch 1/20
150/150 [==============================] - 29s 196ms/step - loss: 0.5626 - acc:
0.7454 - val_loss: 0.5545 - val_acc: 0.7890
Epoch 16/20
149/150 [============================>.] - ETA: 0s - loss: 0.5566 - acc:
0.7634Epoch 1/20
150/150 [==============================] - 25s 165ms/step - loss: 0.5570 - acc:
0.7634 - val_loss: 0.5281 - val_acc: 0.7730
Epoch 17/20
149/150 [============================>.] - ETA: 0s - loss: 0.5185 - acc:
0.7680Epoch 1/20
150/150 [==============================] - 25s 165ms/step - loss: 0.5176 - acc:
0.7679 - val_loss: 0.5107 - val_acc: 0.7890
Epoch 18/20
149/150 [============================>.] - ETA: 0s - loss: 0.5708 - acc:
0.7182Epoch 1/20
150/150 [==============================] - 24s 163ms/step - loss: 0.5722 - acc:
0.7186 - val_loss: 0.5494 - val_acc: 0.7610
Epoch 19/20
149/150 [============================>.] - ETA: 0s - loss: 0.4888 - acc:
0.7983Epoch 1/20
150/150 [==============================] - 25s 164ms/step - loss: 0.4884 - acc:
0.7980 - val_loss: 0.5143 - val_acc: 0.7460
Epoch 20/20
149/150 [============================>.] - ETA: 0s - loss: 0.4652 - acc:
0.7808Epoch 1/20
150/150 [==============================] - 24s 162ms/step - loss: 0.4656 - acc:
0.7809 - val_loss: 0.4766 - val_acc: 0.7940
```

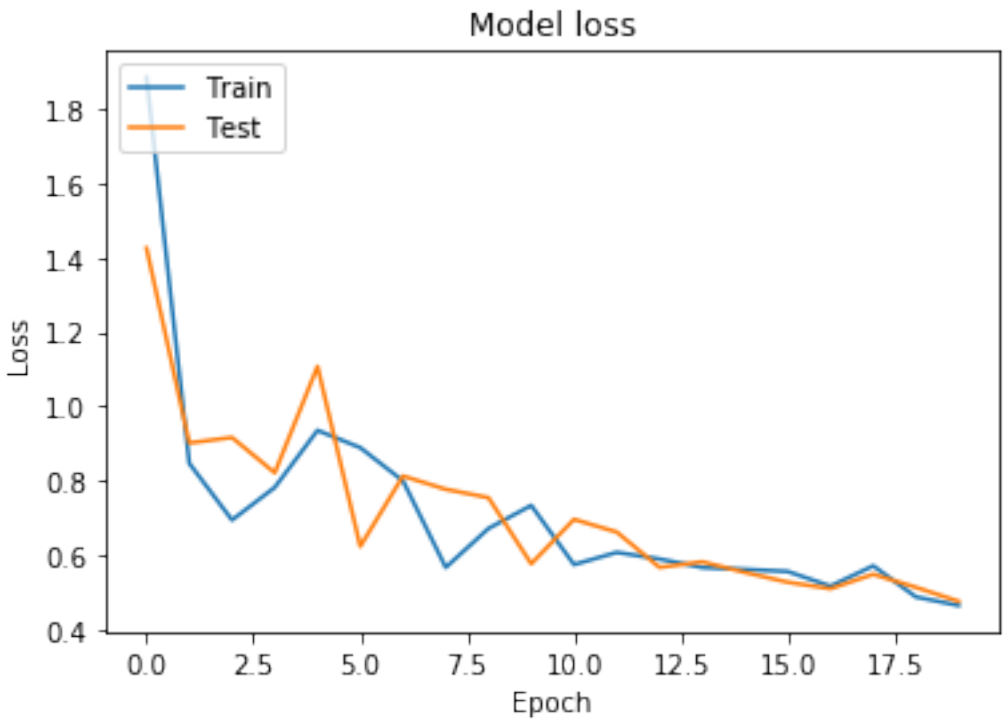
[12]:
```
model.save('/home/user/models/simple/simple_two_class.h5')
```

[13]:
```
    # Plot training & validation accuracy values
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()

    # Plot training & validation loss values
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()
```




Model loss
Train
Test
Loss
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epoch

[ ]: