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
```

[16]:
```
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')
])

opt = optimizers.SGD(learning_rate=0.001, momentum=0.0, nesterov=False)
model.compile(optimizer=opt,
              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: 0.5817 - acc:
0.6580Epoch 1/20
150/150 [==============================] - 26s 172ms/step - loss: 0.5817 - acc:
0.6585 - val_loss: 0.5290 - val_acc: 0.7440
Epoch 2/20
149/150 [============================>.] - ETA: 0s - loss: 0.5193 - acc:
0.7552Epoch 1/20
150/150 [==============================] - 24s 161ms/step - loss: 0.5185 - acc:
0.7551 - val_loss: 0.5260 - val_acc: 0.7390
Epoch 3/20
149/150 [============================>.] - ETA: 0s - loss: 0.5038 - acc:
```

```
0.7542Epoch 1/20
150/150 [==============================] - 24s 160ms/step - loss: 0.5034 - acc:
0.7542 - val_loss: 0.4972 - val_acc: 0.7550
Epoch 4/20
149/150 [============================>.] - ETA: 0s - loss: 0.4834 - acc:
0.7587Epoch 1/20
150/150 [==============================] - 24s 161ms/step - loss: 0.4827 - acc:
0.7588 - val_loss: 0.4925 - val_acc: 0.7510
Epoch 5/20
149/150 [============================>.] - ETA: 0s - loss: 0.4645 - acc:
0.7529Epoch 1/20
150/150 [==============================] - 24s 159ms/step - loss: 0.4640 - acc:
0.7531 - val_loss: 0.4621 - val_acc: 0.7790
Epoch 6/20
149/150 [============================>.] - ETA: 0s - loss: 0.4625 - acc:
0.7665Epoch 1/20
150/150 [==============================] - 24s 161ms/step - loss: 0.4629 - acc:
0.7666 - val_loss: 0.4576 - val_acc: 0.7890
Epoch 7/20
149/150 [============================>.] - ETA: 0s - loss: 0.4467 - acc:
0.7909Epoch 1/20
150/150 [==============================] - 24s 160ms/step - loss: 0.4468 - acc:
0.7909 - val_loss: 0.4908 - val_acc: 0.7430
Epoch 8/20
149/150 [============================>.] - ETA: 0s - loss: 0.4437 - acc:
0.7955Epoch 1/20
150/150 [==============================] - 24s 162ms/step - loss: 0.4430 - acc:
0.7954 - val_loss: 0.4389 - val_acc: 0.8020
Epoch 9/20
149/150 [============================>.] - ETA: 0s - loss: 0.4262 - acc:
0.8122Epoch 1/20
150/150 [==============================] - 24s 161ms/step - loss: 0.4279 - acc:
0.8122 - val_loss: 0.4404 - val_acc: 0.7990
Epoch 10/20
149/150 [============================>.] - ETA: 0s - loss: 0.4313 - acc:
0.8011Epoch 1/20
150/150 [==============================] - 24s 163ms/step - loss: 0.4315 - acc:
0.8011 - val_loss: 0.4508 - val_acc: 0.7760
Epoch 11/20
149/150 [============================>.] - ETA: 0s - loss: 0.4213 - acc:
0.8132Epoch 1/20
150/150 [==============================] - 25s 165ms/step - loss: 0.4223 - acc:
0.8130 - val_loss: 0.4731 - val_acc: 0.7650
Epoch 12/20
149/150 [============================>.] - ETA: 0s - loss: 0.4188 - acc:
0.7985Epoch 1/20
150/150 [==============================] - 24s 163ms/step - loss: 0.4180 - acc:
0.7986 - val_loss: 0.4707 - val_acc: 0.7630
```

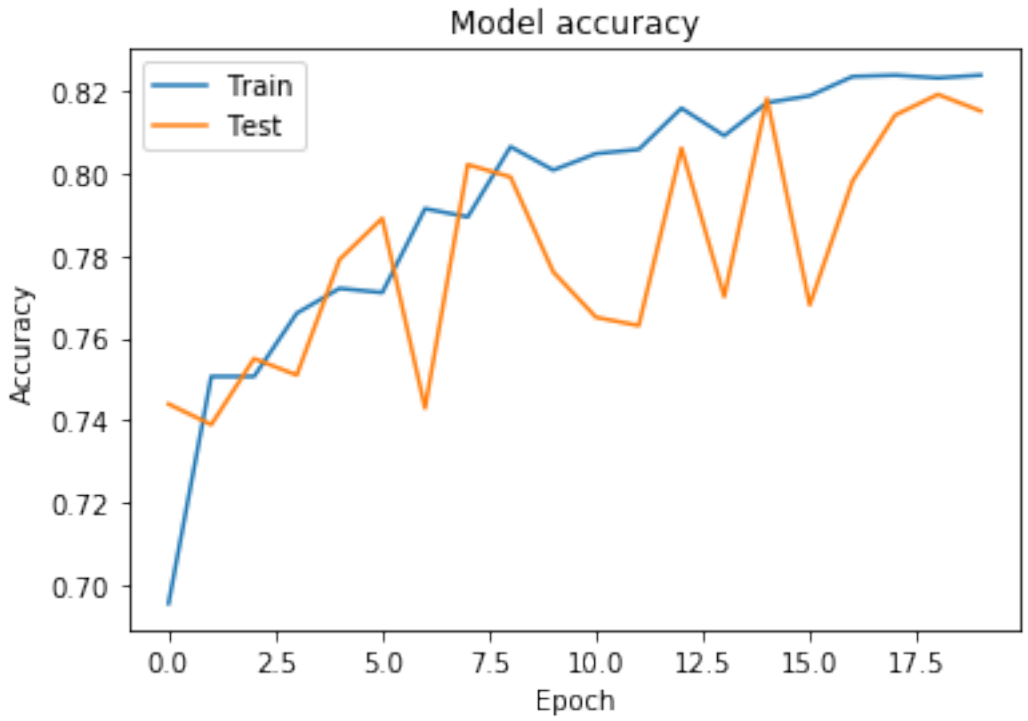
```
Epoch 13/20
149/150 [============================>.] - ETA: 0s - loss: 0.4088 - acc:
0.8202Epoch 1/20
150/150 [==============================] - 24s 161ms/step - loss: 0.4088 - acc:
0.8202 - val_loss: 0.4291 - val_acc: 0.8060
Epoch 14/20
149/150 [============================>.] - ETA: 0s - loss: 0.4130 - acc:
0.8128Epoch 1/20
150/150 [==============================] - 24s 161ms/step - loss: 0.4120 - acc:
0.8127 - val_loss: 0.4541 - val_acc: 0.7700
Epoch 15/20
149/150 [============================>.] - ETA: 0s - loss: 0.4064 - acc:
0.8273Epoch 1/20
150/150 [==============================] - 24s 161ms/step - loss: 0.4057 - acc:
0.8271 - val_loss: 0.4235 - val_acc: 0.8180
Epoch 16/20
149/150 [============================>.] - ETA: 0s - loss: 0.4013 - acc:
0.8248Epoch 1/20
150/150 [==============================] - 24s 161ms/step - loss: 0.4007 - acc:
0.8247 - val_loss: 0.4489 - val_acc: 0.7680
Epoch 17/20
149/150 [============================>.] - ETA: 0s - loss: 0.3940 - acc:
0.8291Epoch 1/20
150/150 [==============================] - 24s 162ms/step - loss: 0.3944 - acc:
0.8290 - val_loss: 0.4474 - val_acc: 0.7980
Epoch 18/20
149/150 [============================>.] - ETA: 0s - loss: 0.3918 - acc:
0.8173Epoch 1/20
150/150 [==============================] - 25s 164ms/step - loss: 0.3921 - acc:
0.8174 - val_loss: 0.4221 - val_acc: 0.8140
Epoch 19/20
149/150 [============================>.] - ETA: 0s - loss: 0.3858 - acc:
0.8249Epoch 1/20
150/150 [==============================] - 24s 162ms/step - loss: 0.3866 - acc:
0.8249 - val_loss: 0.4150 - val_acc: 0.8190
Epoch 20/20
149/150 [============================>.] - ETA: 0s - loss: 0.3937 - acc:
0.8151Epoch 1/20
150/150 [==============================] - 24s 161ms/step - loss: 0.3929 - acc:
0.8152 - val_loss: 0.4129 - val_acc: 0.8150
```

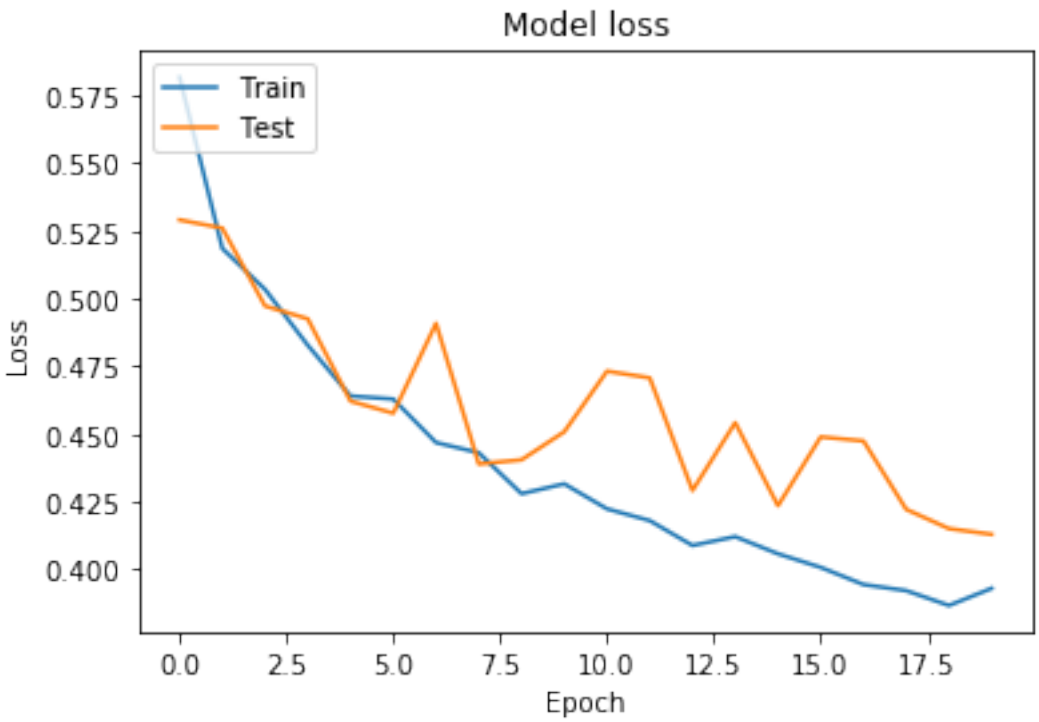
[17]:
```
model.save('/home/user/models/simple/simple_two_class.h5')
```

[18]:
```
    # 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
0.575
0.550
0.525
0.500
0.475
0.450
0.425
0.400
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epoch

[ ]: