Kat Week 4

February 20, 2020

Peer Review Final Assignment

0.1 Introduction

In this lab, you will build an image classifier using the VGG16 pre-trained model, and you will evaluate it and compare its performance to the model we built in the last module using the ResNet50 pre-trained model. Good luck!

0.2 Table of Contents

0.3 Download Data

Use the wget command to download the data for this assignment from here: https://s3-api.us-geo.objectstorage.softlayer.net/cf-courses-data/CognitiveClass/DL0321EN/data/concrete_data_week4.zip

Use the following cells to download the data.

[ ]:
```
!wget https://s3-api.us-geo.objectstorage.softlayer.net/cf-courses-data/
↪CognitiveClass/DL0321EN/data/concrete_data_week4.zip
```

[ ]:
```
!unzip concrete_data_week4.zip
```

After you unzip the data, you fill find the data has already been divided into a train, validation, and test sets.

0.4 Part 1

In this part, you will design a classifier using the VGG16 pre-trained model. Just like the ResNet50 model, you can import the model VGG16 from keras.applications.

You will essentially build your classifier as follows: 1. Import libraries, modules, and packages you will need. Make sure to import the *preprocess_input* function from keras.applications.vgg16. 2. Use a batch size of 100 images for both training and validation. 3. Construct an ImageDataGenerator for the training set and another one for the validation set. VGG16 was originally trained on 224

× 224 images, so make sure to address that when defining the ImageDataGenerator instances. 4. Create a sequential model using Keras. Add VGG16 model to it and dense layer. 5. Compile the mode using the adam optimizer and the categorical_crossentropy loss function. 6. Fit the model on the augmented data using the ImageDataGenerators.

Use the following cells to create your classifier.

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

[ ]:
```
import keras
from keras.models import Sequential
from keras.layers import Dense
```

[ ]:
```
from keras.applications import VGG16
from keras.applications.vgg16 import preprocess_input
```

[ ]:
```
num_classes = 2
image_resize = 224
batch_size_training = 100
batch_size_validation = 100
```

[ ]:
```
data_generator = ImageDataGenerator(
    preprocessing_function=preprocess_input,
)
```

[ ]:
```
train_generator = data_generator.flow_from_directory(
    'concrete_data_week4/train',
    target_size=(image_resize, image_resize),
    batch_size=batch_size_training,
    class_mode='categorical')
```

[ ]:
```
validation_generator = data_generator.flow_from_directory(
    'concrete_data_week4/valid',
    target_size=(image_resize, image_resize),
    batch_size=batch_size_validation,
    class_mode='categorical')
```

[ ]:
```
model = Sequential()
```

[ ]:
```
model.add(VGG16(
    include_top=False,
    pooling='avg',
    weights='imagenet',
    ))
```

[ ]:
```
model.add(Dense(1, activation='softmax'))
```

[ ]:
```
model.layers[0].trainable = False
```

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

[ ]:
```
model.compile(optimizer='adam', loss='categorical_crossentropy',␣
↪metrics=['accuracy'])
```

[ ]:
```
steps_per_epoch_training = len(train_generator)
steps_per_epoch_validation = len(validation_generator)
num_epochs = 2
```

[ ]:
```
fit_history = model.fit_generator(
    train_generator,
    steps_per_epoch=steps_per_epoch_training,
    epochs=num_epochs,
    validation_data=validation_generator,
    validation_steps=steps_per_epoch_validation,
    verbose=1,
)
```

0.5 Part 2

In this part, you will evaluate your deep learning models on a test data. For this part, you will need to do the following:

1. Load your saved model that was built using the ResNet50 model.
2. Construct an ImageDataGenerator for the test set. For this ImageDataGenerator instance, you only need to pass the directory of the test images, target size, and the **shuffle** parameter and set it to False.
3. Use the **evaluate_generator** method to evaluate your models on the test data, by passing the above ImageDataGenerator as an argument. You can learn more about **evaluate_generator** here.
4. Print the performance of the classifier using the VGG16 pre-trained model.
5. Print the performance of the classifier using the ResNet pre-trained model.

Use the following cells to evaluate your models.

[ ]:
```
from keras.models import load_model
```

[ ]:
```
model50 = load_model('classifier_resnet_model.h5')
```

[ ]:
```
test_generator = data_generator.flow_from_directory(
    'concrete_data_week4/test',
    target_size=(image_resize, image_resize),
    shuffle=False)
```

[ ]:
```
scoreVGG16=model.evaluate(test_generator, batch_size_validation, verbose=1)
```

[ ]:
```
scoreResNet50=model50.evaluate(test_generator, batch_size_validation, verbose=1)
```

0.6 Part 3

In this model, you will predict whether the images in the test data are images of cracked concrete or not. You will do the following:

1. Use the **predict_generator** method to predict the class of the images in the test data, by passing the test data ImageDataGenerator instance defined in the previous part as an argument. You can learn more about the **predict_generator** method here.
2. Report the class predictions of the first five images in the test set. You should print something list this:

Positive

Negative

Positive

Positive

Negative

Use the following cells to make your predictions.

[ ]:
```
ResNet50_predict = resnet_model.predict_generator(
    test_generator,
    batch_size_validation,
    verbose=1
)
```

[ ]:
```
VGG16_predict = vgg_model.predict_generator(
    test_generator,
    batch_size_validation,
    verbose=1
)
```

[ ]:
```
ResNet50_classes = ResNet50_predict.argmax(axis=-1)
for i in range(5):
    print('Positive' if ResNet50_classes[i] else 'Negative')
```

[ ]:
```
VGG16_classes = VGG16_predict.argmax(axis=-1)
for i in range(5):
    print('Positive' if VGG16_classes[i] else 'Negative')
```

0.6.1 Thank you for completing this lab!

This notebook was created by Alex Aklson.

This notebook is part of a course on **Coursera** called *AI Capstone Project with Deep Learning*. If you accessed this notebook outside the course, you can take this course online by clicking here.