HW6: Dog and Cat Classification
Due on June 14, 2020 at 11:59 PM

Instructions:

- **Answer the following questions and submit a report (word, & pdf) including codes and screenshots of relevant results, as well as your source code (py or ipynb) on iCampus.**
- **Unzip the "cats_and_dogs_filtered.zip" file to access the datasets; the unzipped "cats_and_dogs_filtered" directory contains subdirectories named "train" (1,000 cat images + 1,000 dog images), "validation" (500 cat images + 500 dog images) and "test" (empty). You will have to upload your own photos in the "your_picture" directory in the "test" directory to answer the questions.**

Given cat and dog images, use the VGG16 model to develop a binary image classifier. At the beginning of your code, set the random seed as your student ID, i.e.,

```
import numpy as np
studentID = 2019711744
np.random.seed(studentID)
```

- **Data Pre-processing**

  1. Training set: 1,000 cat images, 1,000 dog images
  2. Validation set: 500 cat images, 500 dog images
  3. Use `ImageDataGenerator()` and `.flow_from_dataframe()` (https://keras.io/preprocessing/image/) to feed input images.

     Apply data augmentation to the training set to improve the performance.
     * Augmentation options: `rescale=1./255, rotation_range=20, width_shift_range=0.1, height_shift_range=0.1, shear_range=0.1, zoom_range=0.1, horizontal_flip=True, fill_mode='nearest'`

- **Binary Classification Model**

  1. Sequential + Pre-trained VGG16 (except Dense Layer) + Add 2 Dense Layers.
  2. In the pre-trained VGG16 model, use Convolutional and Pooling layers, except for the

last Dense layer (Fully Connected layer). Use the weights from ImageNet parameters where the input image size is (150,150,3).

3. In the first Dense layer, the output size is 256 with the ReLU activation function.
4. In the second Dense layer, the output size is 1 with a sigmoid activation function.
5. During training, use the pre-trained VGG16 model with `trainable = False` to freeze the weights
6. Print the model summary using `.summary()` to show the overall model architecture.

- **Model Compiling**

1. Use binary cross entropy as the loss function
2. Start with a learning rate of 2e-5

- **Model Training**

1. Train your model using `fit_generator()`. Select proper parameters for `steps_per_epoch` and `validation_steps`. `steps_per_epoch` is the total number of steps (batches or samples) before declaring one epoch finished and starting the next epoch; `validation_steps` is the total number of steps (batches or samples) to draw before stopping when performing validation at the end of every epoch (`validation_data` required).
2. Use a batch size of 20 and 30 epochs.

Questions

Question 1. Plot the training accuracy and the validation accuracy for each epoch on a graph, where the X-axis represents the number of epochs and Y-axis represents the accuracy values.

Question 2. Plot the training loss and the validation loss for each epoch on a graph, where the X-axis represents the number of epochs and Y-axis represents the loss values.

Question 3. Use two different optimizers of your choice for your model and show the corresponding accuracy values.

Question 4. Use five different learning rate values of your choice and show the corresponding accuracy values.

Question 5. Propose and apply any method to improve the model performance (e.g., different

model architecture, use of additional training data, etc.). Has the model accuracy increased or decreased and by how much?

Question 6. Use 2 face images of yourself and show what they are classified as. What are the probabilities of each being a dog/cat?

Question 7. Theoretically, how should your dog/cat classifier perform when it is fed with human face images? What is the correct way to approach the task of classifying human face images? Explain in your own words.

Question 8. Fix your code to properly handle the issue of human face image classification in your dog/cat classifier.

Question 9. Is the fixed solution a generalized approach? What happens when fish, bird, or panda images are fed into your classifier in Question 8?

Question 10. How would you handle the different cases presented in Question 9 (fish, bird, or panda images)?