Assignment 4b

魏家博 (Chia-Po Wei)

Department of Electrical Engineering

National Sun Yat-sen University

Outline

1. Kaggle Dogs and Cats Dataset
    - Preprocess Images
    - Model Fitting with Keras

Steps for Assignment 4b-1

1. Open assignment_4b_template.ipynb
2. Create `dogs_cats.pkl` from train_list.txt and test_list.txt
   1. train_list.txt contains 20000 image names for training.
   2. test_list.txt contains 5000 images names for testing.
   3. The class label is 0 if the image name has cat.
      The class label is 1 if the image name has dog.
   4. Create X_train, Y_train, X_test, Y_test
      X_train.shape is of shape (num_X, 48, 48, 3)
      Y_train.shape is of shape (num_X,)
      X_test.shape is of shape (num_Y, 48, 48, 3)
      Y_test.shape is of shape (num_Y,)
      The dtype of each element is set to np.uint8.
      num_X may be less than 20000 if some files are corrupted.
      num_Y may be less than 5000 if some files are corrupted.
   5. Save X_train, Y_train, X_test, Y_test to `dogs_cats.pkl`

Folder structure

```
+-- current_dir/
|      +-- train/
```

Contents of train_list.txt:

```
dog.0.jpg
cat.0.jpg
dog.1.jpg
cat.1.jpg
```
…

Steps for Assignment 4b-1

Implement the following pseudo code (綠色字體是要實作的部分)

```
def get_data_from_file(train_file):
    train_data = []
    with open(train_file) as fp:
        lines = fp.readlines()
    for k,line in enumerate(lines):
        print('{:6d} /{:6d}'.format(k+1, len(lines)), end='\r')
        obtain img_path from line  (make sure that your img_path is correct)
        try:
            img = cv2.imread(img_path)
            img_resized = cv2.resize(img, (48, 48))
            if line contains cat
                label = 0
            if line contains dog
                label = 1
            train_data.append([img_resized, label])
        except:
            print error message
    return train_data
```

Steps for Assignment 4b-1

- `train_data = get_data_from_file(train_file)`
- `X_train, Y_train = get_image_and_label(train_data)`
- Implement `get_image_and_label()` such that

  `X_train.shape` is of shape (num_X, 48, 48, 3)
  `Y_train.shape` is of shape (num_X,)
  num_X is the length of `train_data`

- Below are part of `assignment4b_1()`

```
train_data = get_data_from_file(train_file)
X_train, Y_train = get_image_and_label(train_data)
test_data = get_data_from_file(test_file)
X_test, Y_test = get_image_and_label(test_data)
save_path = 'dogs_cats.pkl'
print('Saving to', save_path)
data = {}
data['X_train'] = X_train
data['Y_train'] = Y_train
data['X_test'] = X_test
data['Y_test'] = Y_test
pickle.dump(data, open(save_path, 'wb'))
```

Steps for Assignment 4b-2

1. Define a keras model in `build_model()` to have the following summary:

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 48, 48, 3)]       0
 conv2d_3 (Conv2D)           (None, 48, 48, 32)        896
 max_pool_3 (MaxPooling2D)   (None, 24, 24, 32)       0
 conv2d_4 (Conv2D)           (None, 24, 24, 32)        9248
 max_pool_4 (MaxPooling2D)   (None, 12, 12, 32)       0
 conv2d_5 (Conv2D)           (None, 12, 12, 32)        9248
 max_pool_5 (MaxPooling2D)   (None, 6, 6, 32)         0
 flatten_1 (Flatten)         (None, 1152)              0
 dense_1 (Dense)             (None, 2)                 2306
=================================================================
Total params: 21,698
```

- The layer name can be anything, but the layer type and output shape must match those from above.

Steps for Assignment 4b-2

2. Use below to fit your model

```
history = model.fit(X_train, Y_train, epochs=60, batch_size=64,
callbacks=[cp_callback], validation_split=0.1, shuffle=True)
```

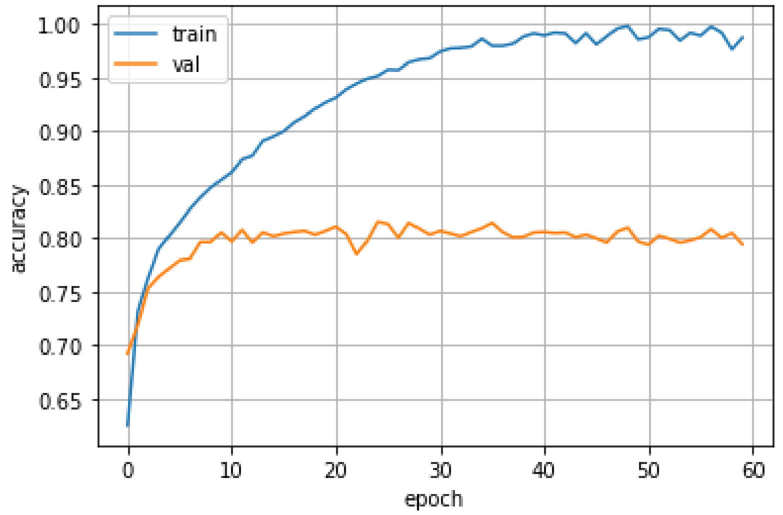
3. Remember to save your training log with jupyter notebook.

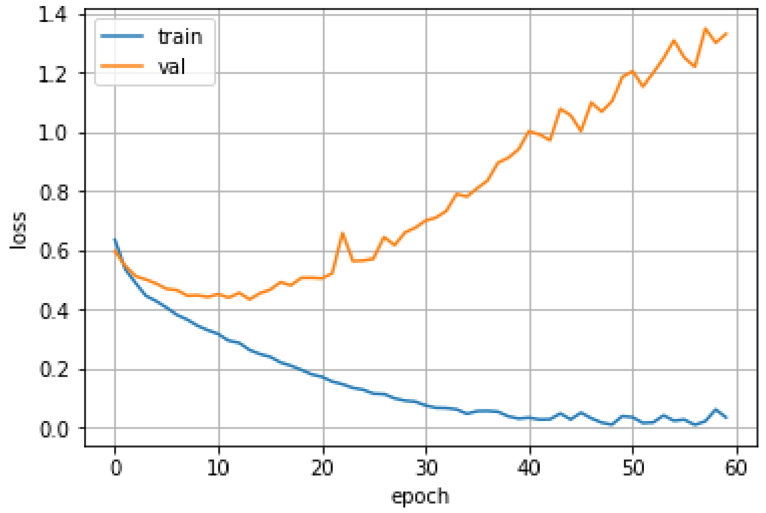
4. After `model.fit()`, display the test accuracy of the trained model on `(X_test, Y_test)`

5. Use `pickle.dump()` to save `history.history` in `'history.pkl'` in binary encodings.

Steps for Assignment 4b-3

1. Use `pickle.load()` to load `'history.pkl'` in the variable `history`.
2. Plot training/validation accuracy vs. epochs as in the left figure below.
3. Plot training/validation loss vs. epochs as in the right figure below.
   - Use `history['accuracy']` and `history['val_accuracy']` to access the training and validation accuracy.
   - Use `history['loss']` and `history['val_loss']` to access the training and validation loss.





- https://machinelearningmastery.com/display-deep-learning-model-training-history-in-keras/

- Assignment 4b-4: Display the test accuracy on `(X_test, Y_test)` for epoch 10, 20, …, 60.
- Question 1: If we set `shuffle=False` during the execution of `model.fit()` in Assignment 4b-2, which was previously set to `shuffle=True`, what impact will this have on the validation accuracy?
- Question 2: In Assignment 4b-4, what epoch yields the highest test accuracy, and what is the reason for this?