

Report for Project 2

MENG Qian 20466688

In this project, it is a classification problem which is used to identify images and the algorithm is Convolutional Neural Network. We need to train a deep model and test on a test set including 5 kinds of flowers. To solve this problem, it is suitable to use keras: the Python deep learning library. Keras is an open source neural network library written in Python. It is capable of running on top of MXNet, DeepLearning4j, Tensorflow, CNTK or Theano, which is highly modular, minimalist and scalable.

Import data

scipy.misc.imread (Read an image from a file as an array)

```
# import data
my_train = pd.read_table('train.txt', sep=' ', header=None)
my_train.columns = ['path', 'label']
my_val = pd.read_table('val.txt', sep=' ', header=None)
my_val.columns = ['path', 'label']

# read images and their labels
image_matrix = imread(my_train.iloc[0, :].path)
image_label = my_train.iloc[0, 1]
```

Data preprocessing

It is necessary to preprocess the data to form that we need. To fit the model, we need to resize the images according to the parameters in the model which we use.

```
# resize the images
my_data = imresize(image_matrix, (224, 224))
height, width, depth = my_data.shape
num_train = len(my_train)
num_val = len(my_val)
num_classes = len(np.unique(my_train.label.values))
my_train_X = np.zeros((num_train, height, width, depth))
my_val_X = np.zeros((num_val, height, width, depth))

for i in range(num_train):
    img = imread(my_train.iloc[i, :].path)
    img = imresize(img, (224, 224))
    my_train_X[i, :, :, :] = img

for i in range(num_val):
    img = imread(my_val.iloc[i, :].path)
    img = imresize(img, (224, 224))
    my_val_X[i, :, :, :] = img

my_train_Y = my_train.label.values.reshape((-1, 1))
my_val_Y = my_val.label.values.reshape((-1, 1))
my_train_Y = np_utils.to_categorical(my_train_Y, num_classes).reshape((-1, num_classes))
my_val_Y = np_utils.to_categorical(my_val_Y, num_classes).reshape((-1, num_classes))
```

Training model

I select ResNet50 as my training model and adjust the parameters.

```
from keras.applications import ResNet50

base_model = ResNet50(include_top=False, weights='imagenet', input_tensor=Input((224, 224, 3)))

x = base_model.output
x = Flatten(name='flatten')(x)
x = Dropout(0.6)(x)
x = Dense(5, activation='softmax', name='predictions')(x)

my_model = Model(inputs=base_model.input, output=x)
layers = base_model.layers
for layer in layers:
    layer.trainable = False
my_model.compile(optimizer='Adagrad',
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])
my_model.fit(my_train_X, my_train_Y, batch_size=64, nb_epoch=6, validation_split=0.2)
result = my_model.predict(my_val_X)
my_model.save('my_model.h5')
```

Test the test data

```
import numpy as np
import pandas as pd
from scipy.misc import imresize
from scipy.misc import imread
from keras.models import load_model

test = pd.read_table('val.txt', header=None, sep=' ', encoding='gb2312')
test.columns = ['path', 'label']

image_matrix = imread(test.iloc[0, :].path)
my_image = imresize(image_matrix, (224, 224))
height, width, depth = my_image.shape
num_test = len(test)
X_test = np.zeros((num_test, height, width, depth))

for i in range(num_test):
    image = imread(test.iloc[i, :].path)
    image = imresize(image_matrix, (224, 224))
    X_test[i, :, :, :] = image

my_model = load_model('my_model.h5')
my_predict = my_model.predict(X_test)

result = np.argmax(my_predict, axis=1)
print(result)
np.savetxt("test_result.csv", result, delimiter=",")
```