Project 2 for Midterm

The whole project is about flower classification.

1. **Data Processing**
    a. We use train data to build model, val data to test the model. Then using test data to get the test result.
    b. We have train.txt, val.txt, test.txt, each file has path to **read the image**, write a for loop to read path one by one, then add all the image into a list, and convert the list becomes **an array**.
    c. **Resize** the image size, because the origin image has different size, need become consolidate shape.
    d. These image of the specified size converted to a float range. Because typically 24-bit images are in the range[0,255], and in order to **convert them to the [-1,1],** float range expected by the model divided by 255.
    e. So we get the format to train, **shape** likes (num_samples, row, col, 3)
    f. The train and val files have label, read the label out and add into a list.
    g. Using **one-hot** convert class vectors to binary class matrices.

2. Tool: **Keras** using Tensorflow backend

3. **Build CNN mode**
    a. Add Conv 2D, set filter, kernel size, activation, strides, input size, padding=' same'
    b. Add maxpooling, set pooling size, strides and padding
    c. One conv layer, one maxpooling layer, alternately
    d. Set dropout avoid prevent overfitting
    e. Flatten
    f. Add two Fully connect layer, one using 'relu' , one is 'softmax' , the output is number class.

4. Model.compile

a. set the loss, optimizer and metric. optimizer using Adadelta, An Adaptive Learning Rate Method, the learning rate is changing.

b. fit the model and evaluate.

5. Adjust parameter

a This is the most important part:

batch_size, at first set 128, the become 100, final set 80.

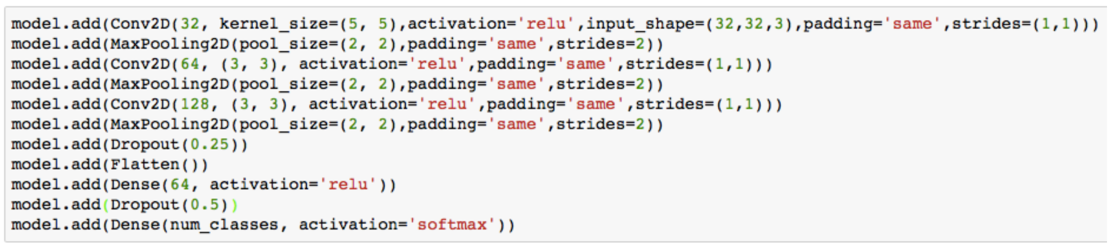
num_classes = 5

epochs = 50, at first 20, then 30, at last 50

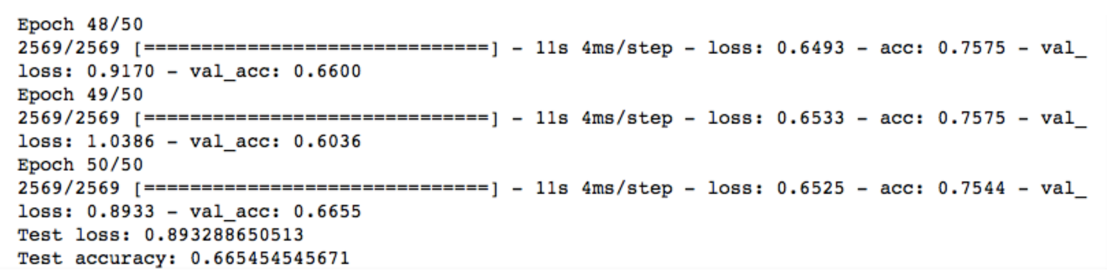
b. Resize the picture, first 128*128, then change to 64*64, and last set to 32*32, because large dimension is not good for accuracy.

c. about conv layer, at first add two layers, then adjust to three conv layers, each conv layer following maxpooling layer.

6. final result



```
model.add(Conv2D(32, kernel_size=(5, 5),activation='relu',input_shape=(32,32,3),padding='same',strides=(1,1)))
model.add(MaxPooling2D(pool_size=(2, 2),padding='same',strides=2))
model.add(Conv2D(64, (3, 3), activation='relu',padding='same',strides=(1,1)))
model.add(MaxPooling2D(pool_size=(2, 2),padding='same',strides=2))
model.add(Conv2D(128, (3, 3), activation='relu',padding='same',strides=(1,1)))
model.add(MaxPooling2D(pool_size=(2, 2),padding='same',strides=2))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))
```



```
Epoch 48/50
2569/2569 [==============================] - 11s 4ms/step - loss: 0.6493 - acc: 0.7575 - val_
loss: 0.9170 - val_acc: 0.6600
Epoch 49/50
2569/2569 [==============================] - 11s 4ms/step - loss: 0.6533 - acc: 0.7575 - val_
loss: 1.0386 - val_acc: 0.6036
Epoch 50/50
2569/2569 [==============================] - 11s 4ms/step - loss: 0.6525 - acc: 0.7544 - val_
loss: 0.8933 - val_acc: 0.6655
Test loss: 0.893288650513
Test accuracy: 0.665454545671
```

The train process is very slow, each time need 3-4 hours training, then adjust parameter. Some time epoch is not enough, loss is still descending, some time overfitting, train accuracy is very high, but val is very low. finally get the result.