MSBD 6000b project2

Li Meng 李萌 20477168

1. Abstract

In this project, I apply CNN on flower images classification task. First I preprocessing the images into same size. Then, models with different layers and kernel sizes have been test to find a better result. Finally, I apply a larger epoch on the model to predict the test label.

2. Data preprocessing

In this section, we transform the input images to the same size, 100*100, by using the skimage.transform function. The smallest image size is 159*143, so for all images, I performs interpolation to down-size them.

All the images are transformed into float32, and the label are represented in one-hot vector.

3. CNN Network

I use a small convnet with few layers and few filters per layer, alongside data augmentation and dropout. The dropout layers help reduce overfitting. A simple stack of 3 convolution layers with a RELU activation and followed by max-pooling layers. On the top of it we stick two fully-connected layers. The structure of my network is as below,

Layer (type)	Output Shape	Param #
conv2d_11 (Conv2D)	(None, 100, 100, 32)	896
activation_17 (Activation)	(None, 100, 100, 32)	0
max_pooling2d_11 (MaxPooling	(None, 50, 50, 32)	0
conv2d_12 (Conv2D)	(None, 50, 50, 64)	18496
activation_18 (Activation)	(None, 50, 50, 64)	0
max_pooling2d_12 (MaxPooling	(None, 25, 25, 64)	0
conv2d_13 (Conv2D)	(None, 25, 25, 64)	36928
activation_19 (Activation)	(None, 25, 25, 64)	0
max_pooling2d_13 (MaxPooling	(None, 12, 12, 64)	0
flatten_4 (Flatten)	(None, 9216)	0
dense_7 (Dense)	(None, 64)	589888
activation_20 (Activation)	(None, 64)	0
dropout_4 (Dropout)	(None, 64)	0
dense_8 (Dense)	(None, 5)	325
activation_21 (Activation)	(None, 5)	0

Total params: 646,533

For the loss, I use categorical_crossentropy, the optimizer is RMSprop.

4. Experiments

epoch	Train_Loss	Train_acc	Val_loss	Val_acc
5	0.2693	0.9288	2.5310	0.6872
20	0.2545	0.9389	2.8752	0.6981
50	0.0962	0.9504	2.0593	0.6152

From the table above we can see clearly that when epoch is 50, the model is overfitting. When epoch=20, we can achieve about 70% accuratcy. So we chose this model to predict the test set.