期末報告
[multiCNN]

學號：r10525069
姓名：林子傑

Outline

Dataset and Targat Introduction

Algorithm Introduction

Code Review

Live Demo(Using InAnalysis)

Conclusion

Reference

Dataset and Target Introduction

旅宿景點（trip_dataset）當中取 632 張當作資料集

500 張 Training、132 張 Testing

圖片分成 4 類

- “text”有說明或導覽文字的圖片
- “food” 有食物的圖片
- “people”有人的圖片
- “landspace” 其他圖片

利用 CNN 演算法得出一個模型能分辨這四類。

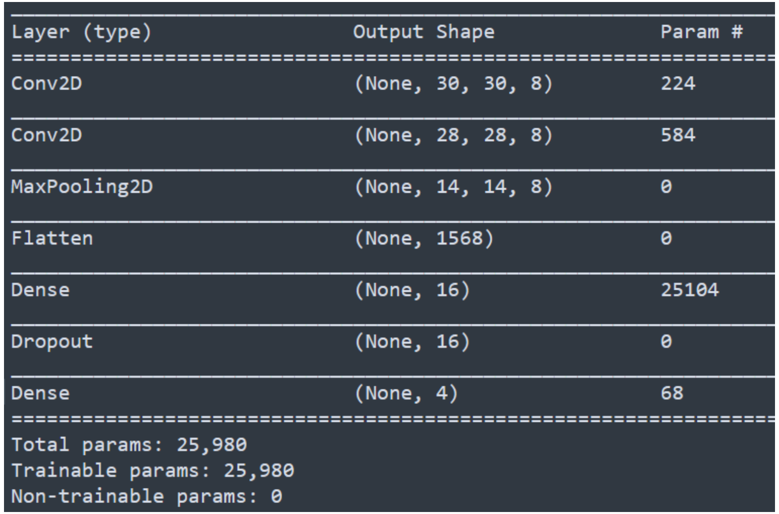
Algorithm Introduction

Convolution -> Convolution -> MaxPooling2D

-> Flatten -> Dense -> Dropout -> Dense

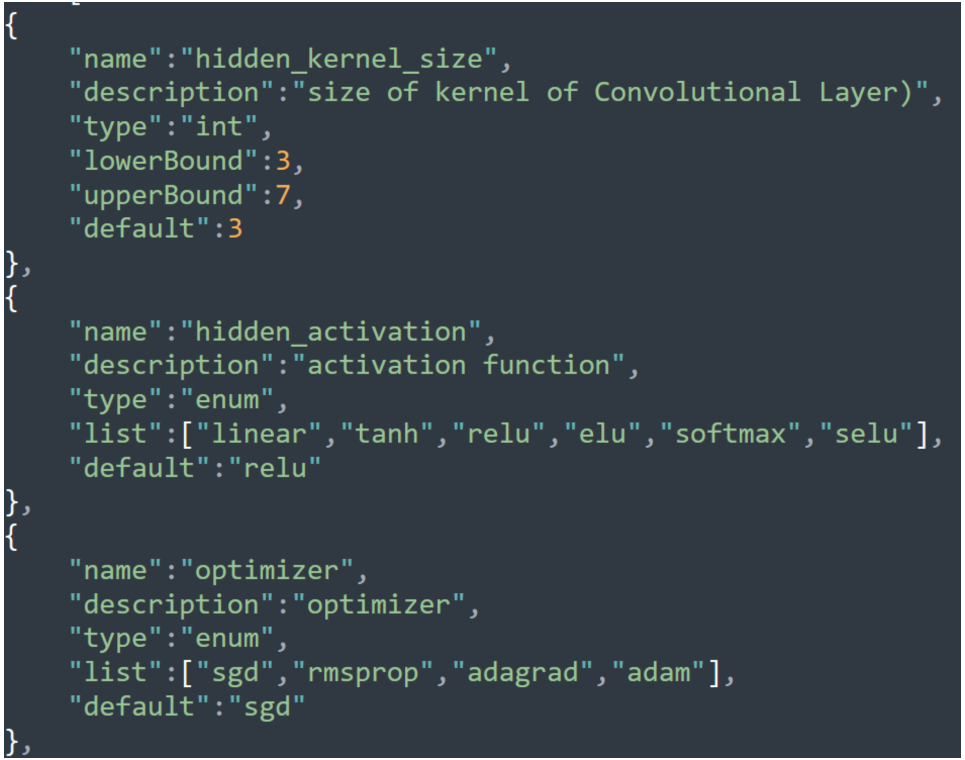
輸入大小-> 32*32*3

輸出 -> sofmax 的值

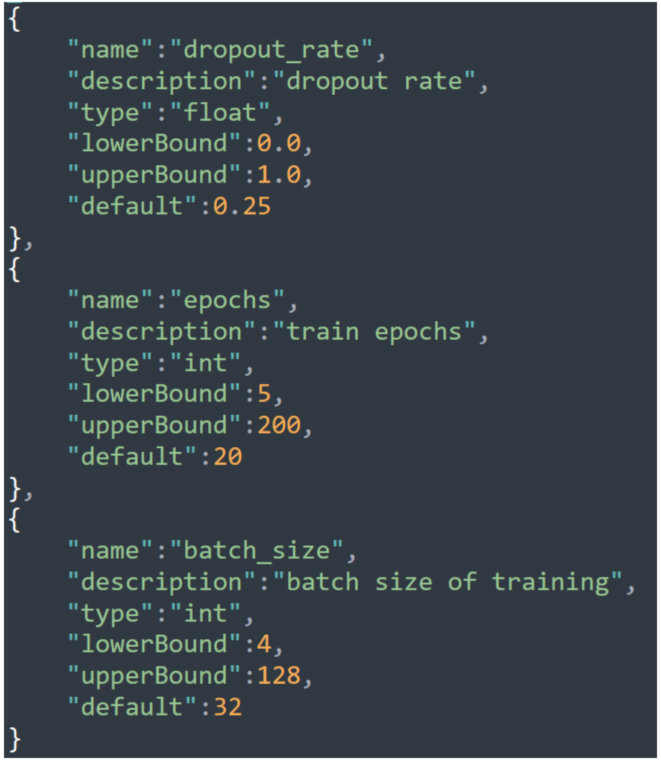


```
_________________________________________________________________
Layer (type)                 Output Shape               Param #
=================================================================
Conv2D                       (None, 30, 30, 8)          224
_________________________________________________________________
Conv2D                       (None, 28, 28, 8)          584
_________________________________________________________________
MaxPooling2D                 (None, 14, 14, 8)          0
_________________________________________________________________
Flatten                      (None, 1568)               0
_________________________________________________________________
Dense                        (None, 16)                 25104
_________________________________________________________________
Dropout                      (None, 16)                 0
_________________________________________________________________
Dense                        (None, 4)                  68
=================================================================
Total params: 25,980
Trainable params: 25,980
Non-trainable params: 0
```

Code Review

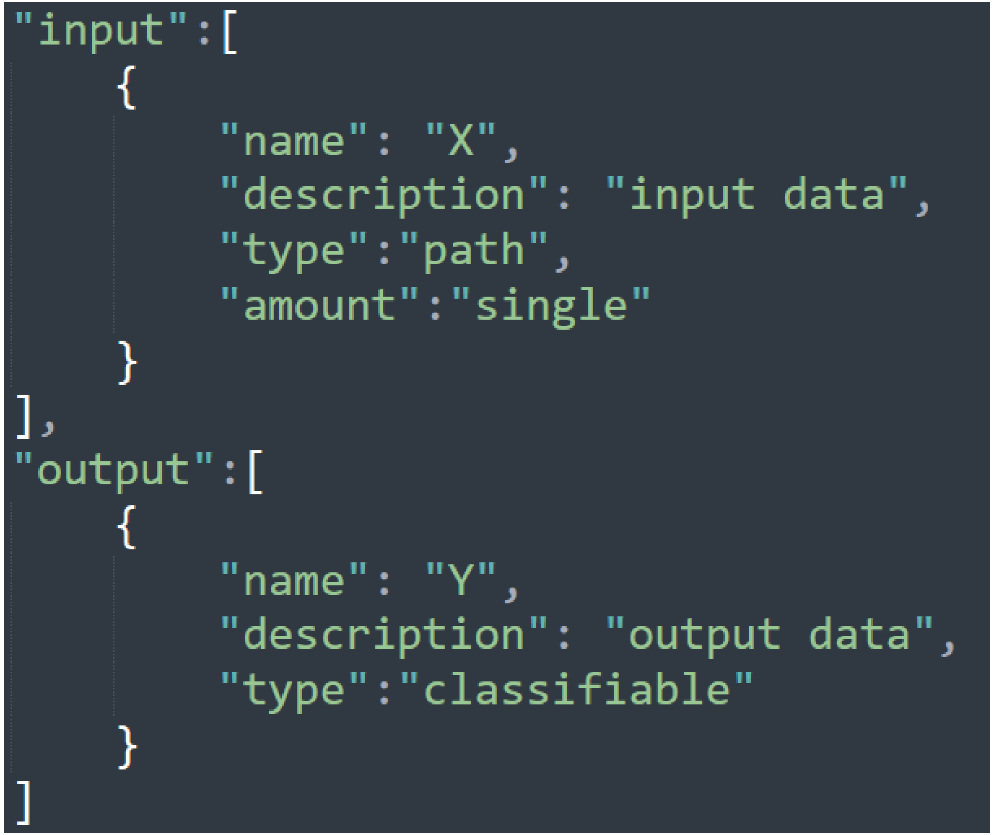


```
{
    "name":"hidden_kernel_size",
    "description":"size of kernel of Convolutional Layer)",
    "type":"int",
    "lowerBound":3,
    "upperBound":7,
    "default":3
},
{
    "name":"hidden_activation",
    "description":"activation function",
    "type":"enum",
    "list":["linear","tanh","relu","elu","softmax","selu"],
    "default":"relu"
},
{
    "name":"optimizer",
    "description":"optimizer",
    "type":"enum",
    "list":["sgd","rmsprop","adagrad","adam"],
    "default":"sgd"
},
```



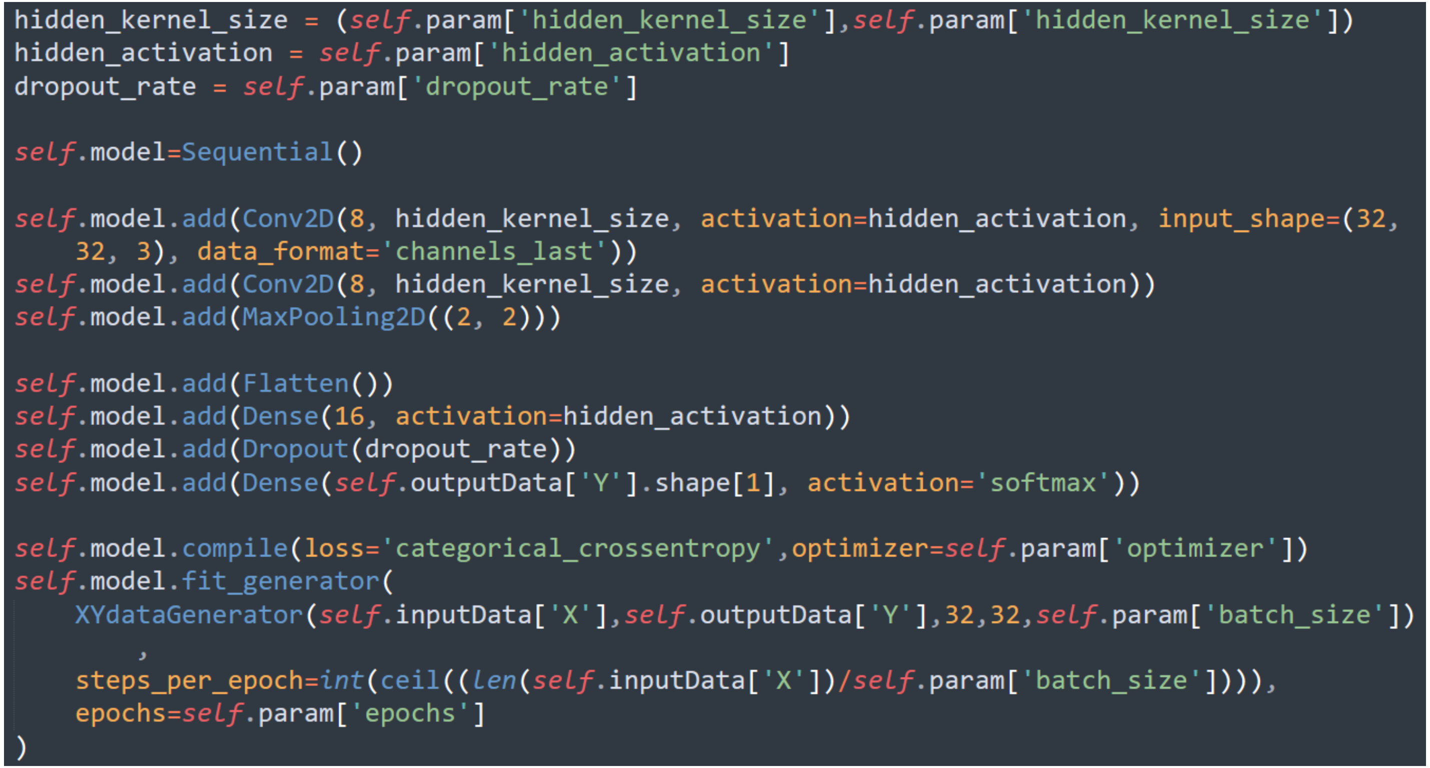
```
{
    "name":"dropout_rate",
    "description":"dropout rate",
    "type":"float",
    "lowerBound":0.0,
    "upperBound":1.0,
    "default":0.25
},
{
    "name":"epochs",
    "description":"train epochs",
    "type":"int",
    "lowerBound":5,
    "upperBound":200,
    "default":20
},
{
    "name":"batch_size",
    "description":"batch size of training",
    "type":"int",
    "lowerBound":4,
    "upperBound":128,
    "default":32
}
```

Code Review



```
"input":[
    {
        "name": "X",
        "description": "input data",
        "type":"path",
        "amount":"single"
    }
],
"output":[
    {
        "name": "Y",
        "description": "output data",
        "type":"classifiable"
    }
]
```

Code Review



```
hidden_kernel_size = (self.param['hidden_kernel_size'],self.param['hidden_kernel_size'])
hidden_activation = self.param['hidden_activation']
dropout_rate = self.param['dropout_rate']

self.model=Sequential()

self.model.add(Conv2D(8, hidden_kernel_size, activation=hidden_activation, input_shape=(32,
    32, 3), data_format='channels_last'))
self.model.add(Conv2D(8, hidden_kernel_size, activation=hidden_activation))
self.model.add(MaxPooling2D((2, 2)))

self.model.add(Flatten())
self.model.add(Dense(16, activation=hidden_activation))
self.model.add(Dropout(dropout_rate))
self.model.add(Dense(self.outputData['Y'].shape[1], activation='softmax'))

self.model.compile(loss='categorical_crossentropy',optimizer=self.param['optimizer'])
self.model.fit_generator(
    XYdataGenerator(self.inputData['X'],self.outputData['Y'],32,32,self.param['batch_size'])
        ,
    steps_per_epoch=int(ceil((len(self.inputData['X'])/self.param['batch_size']))),
    epochs=self.param['epochs']
)
```

Live demo

Conclusion

1. Keras 有助於理解 CNN 架構
2. 參數個數和資料及大小的關係

Reference

1. https://medium.com/@iamvarman/how-to-calculate-the-number-of-parameters-in-the-cnn-5bd55364d7ca

2. https://www.kaggle.com/kasana/image-classification-using-keras-cnn/notebook

3. https://learnopencv.com/image-classification-using-convolutional-neural-networks-in-keras/

4. https://keras.io/api/