**Sample Code & 作業內容**

請參閱作業範例：Day95-CNN\_Pooling\_Padding.ipynb

範例說明：

(1)zero\_pad - feature map 跟原圖的比對

(2)pool\_forward - 前行網路的 pooling layer

作業：

(1) 調整Padding 的設定，看 Pad 後的結果

(2) 調整Pooling 的設定，看 Poolin 後的結果

作業請提交：Day95-CNN\_Pooling\_Padding\_HW.ipynb

[檢視範例](https://ai100-2.cupoy.com/samplecodelist/D95)

**參考資料**

**以 Tensorflow 為例**

# 池化

def max\_pool\_2x2(x):

    return tf.nn.max\_pool(x, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding='SAME')

# 池化跟卷積的情況有點類似

# x 是卷積後,有經過非線性啟動後的圖像,

# ksize 是池化滑動張量

# ksize 的維度[batch, height, width, channels],跟 x 張量相同

# strides [1, 2, 2, 1],與上面對應維度的移動步長

# padding與卷積函數相同,padding='VALID',對原圖像不進行0填充

**延伸閱讀:基於Keras的卷積神經網路（CNN）視覺化**

**連結：**▪[**https://blog.csdn.net/weiwei9363/article/details/79112872**](https://blog.csdn.net/weiwei9363/article/details/79112872)

本文整理自[**Deep Learning with Python**](https://book.douban.com/subject/27038207/)，書本上完整的代碼在 [**這裡的5.4節**](https://github.com/fchollet/deep-learning-with-python-notebooks)，介紹三種視覺化方法：

* **卷積核輸出的視覺化(Visualizing intermediate convnet outputs (intermediate activations)**，即視覺化卷積核經過啟動之後的結果。能夠看到圖像經過卷積之後結果，幫助理解卷積核的作用
* **卷積核的視覺化(Visualizing convnets filters)**，説明我們理解卷積核是如何感受圖像的
* **熱度圖視覺化(Visualizing heatmaps of class activation in an image)**，通過熱度圖，瞭解圖像分類問題中圖像哪些部分起到了關鍵作用，同時可以定位圖像中物體的位置。

**延伸閱讀: CNN for Image 過程說明**

