

## 第六周

第六周的課程內容為捲積神經網路(Convolution Neural Network, CNN)。

### 1. 稀疏連接

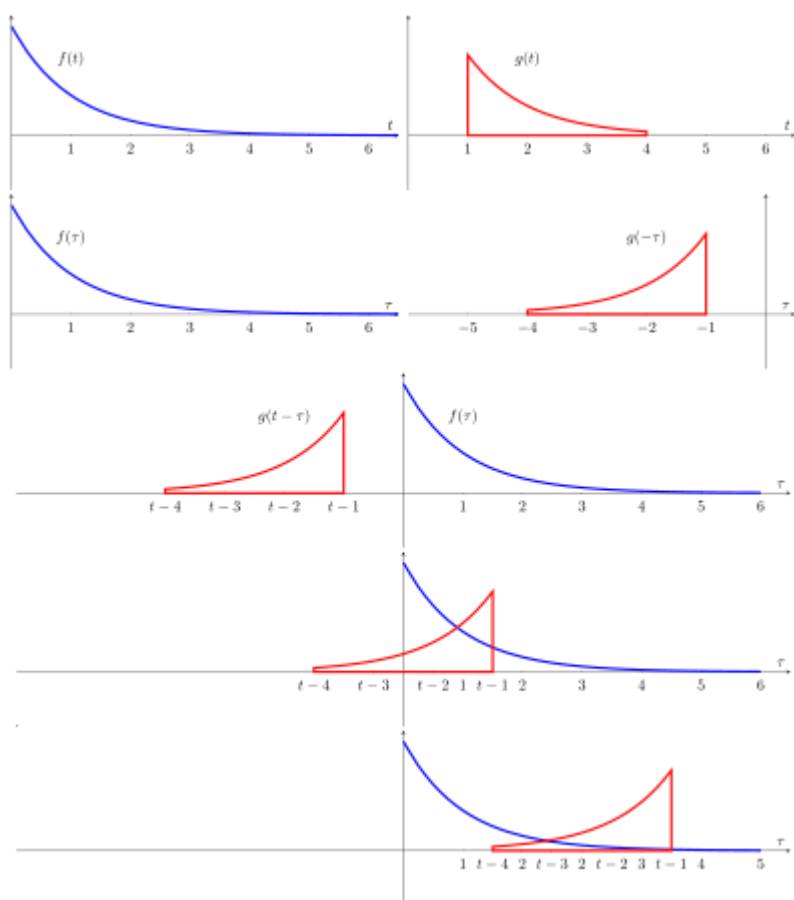
一張圖當中，相鄰的像素會比較遠的像素有關聯，稱為**區域特徵**。例如：一張圖中的邊、角、點等。因此，當神經網路學習時不需以整張圖去學習，只需要以部分有關聯的區域來學習此種方法稱為稀疏連接。如此可以增加準確性、節省運算資源，因此，稀疏連接也屬於一種 Regularization。

### 2. 參數共享

某些區域特徵會在一張圖中重複出現(如點、線、角等)。在訓練神經網路時，若遇到重複出現的特徵，便把之前學習到的參數拿出來使用，不須重新學習。

### 3. 捲積(Convolution)

$s(t) = \int x(a) \times w(t-a) da$ ，其中， $s(t)$ 、 $x(a)$ 、 $w(t-a)$ 皆為訊號。可以理解為將兩個訊號做平移處理後，找出他們之間的積分做為一個新的訊號。卷積屬於一種線性運算。



1. 有兩個函數  $f(a)$  和  $g(a)$

2. 將  $g(a)$  沿縱軸翻轉，變成  $g(-a)$ ； $f(a)$  則不做處理。

3. 將  $g(-a)$  向右平移  $t$ ，變成  $g(t-a)$ 。

4. 因  $t$  並非常數( $t$  是時間變數)。因此可透過調整  $t$  的值，言  $t$  軸滑動( $t$  軸為滑軌，又稱 kernel)

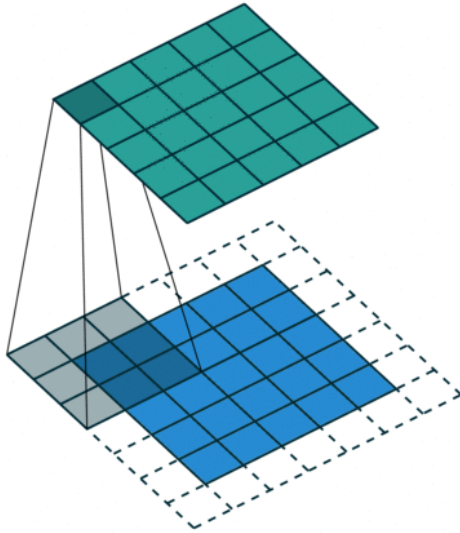
以上的捲積為一維捲積，而一般捲積層用來處理圖片的則常用二維捲積。

二維捲積(以一張圖片為例)：

①輸入為一個矩陣，包含了：[圖片數量(N), 高度(H), 寬度(W), RGB 色彩(C)]。

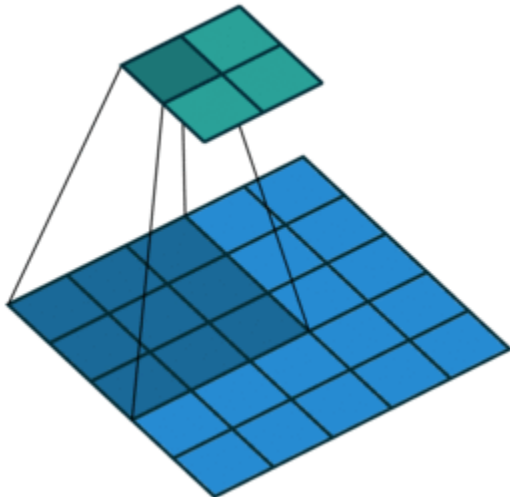
②Kernal：[Kernal 大小(K), Kernal 大小(K), RGB 色彩(C)]。其中，RGB 色彩的 Channel(C)值和輸入一樣。Kernal 大小決定一次取的範圍，此大小權重稱為「捲積核」。

③Padding：如果 Kernal 會超出邊界，將超出的範圍補上稱為 Padding。



圖中的白色部分為補上 1 的邊界，並讓 3×3 的 Kernal 在單次步長為一時可以正常讀取。

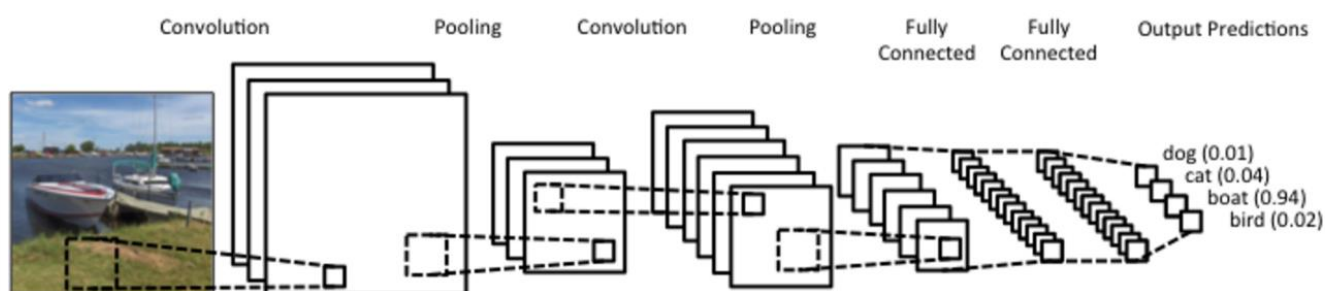
④Stride：一次移動多少步數，若不想 Padding，也可改變單次移動步數。



將原本的步長 1 改成二，便不用補邊界。

△另外，捲積和反捲積互為反操作(可理解為捲積是矩陣相乘，反捲積則是捲積的反矩陣)

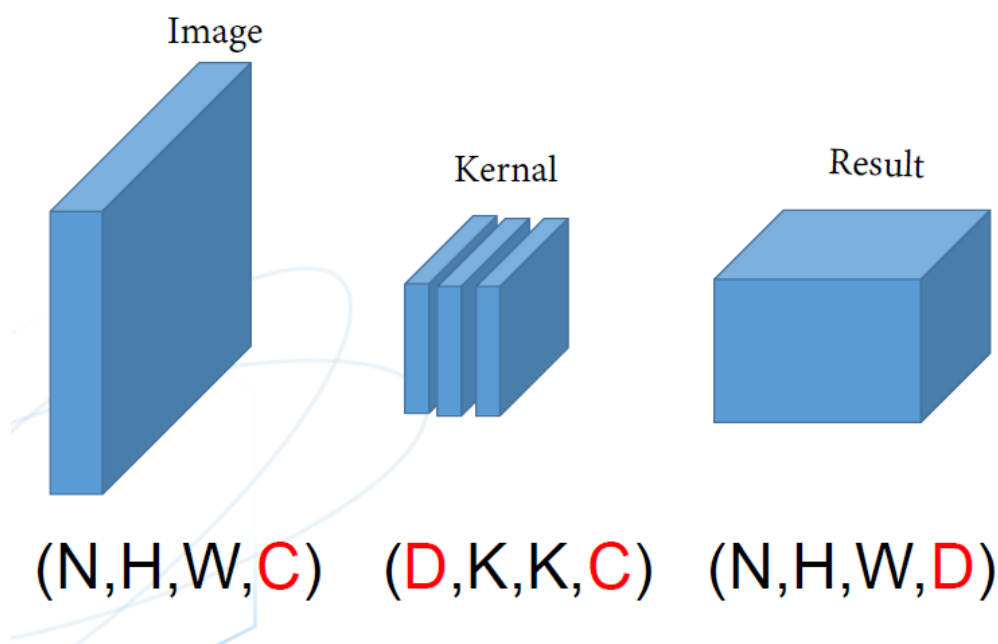
#### 4. 捲積神經網路架構



可以看到，一張圖片需經過捲積層(Convolution layer)和池化層(Pooling layer)的處理後才進到全連結層。

##### ① 卷積層

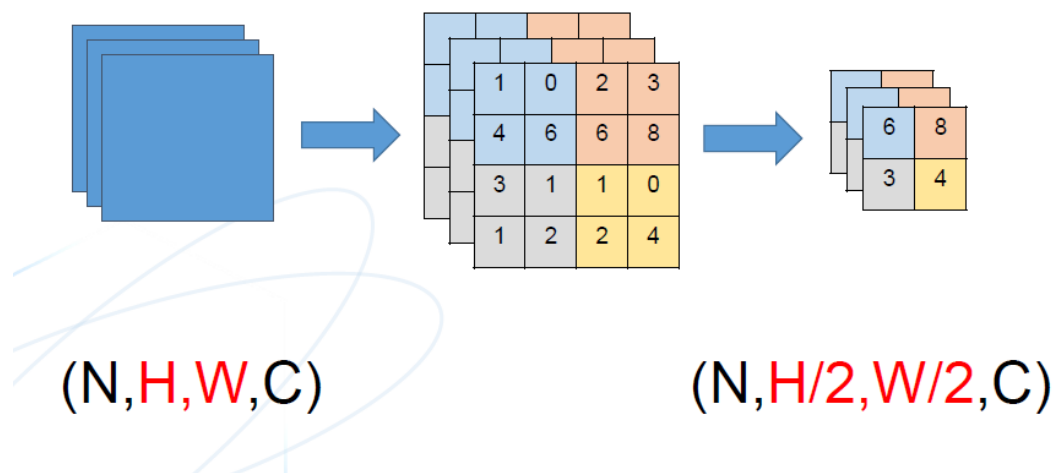
而之前提過，Kernal 和圖片皆可用矩陣去包含資訊，那在經過捲積層後，矩陣中的數值會有什麼變化？



如上圖，有  $N$  張圖片和  $D$  個 Kernal，Channel(這張圖中可理解為厚度)值皆為一樣。將圖片經過這些 Kernal 後，結果的 Channel 值變成了  $D$ (因為是經過了  $D$  張 Kernal 的處理)。

Kernal 每次取的範圍便是區域特徵。另外，在訓練的過程中時常會出現冗於特徵(同一種特徵，但角度不同。例如：45 度角的線和 60 度角的線)

##### ② 池化層



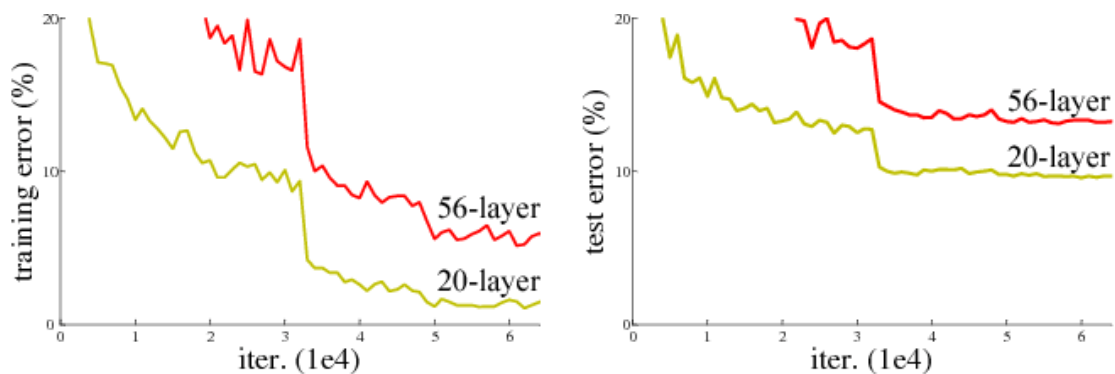
如上圖，池化層的作用是在將一個面分割後，取區域的最大值。

優點：

1. 可見少計算量
2. 過濾一個區域的最強信號。只看該區域是否有該特徵，而忽略了該特徵的確切位置。如此，可以減少神經網路訓練時受到位置影響，增加特徵的不變性。
3. 配合捲積層，可以使資料的維度改變(由寬淺變窄深)來增加神經元的感受區域(因層數增加，分工更細，結果更精準)，進而使表徵能力增加。

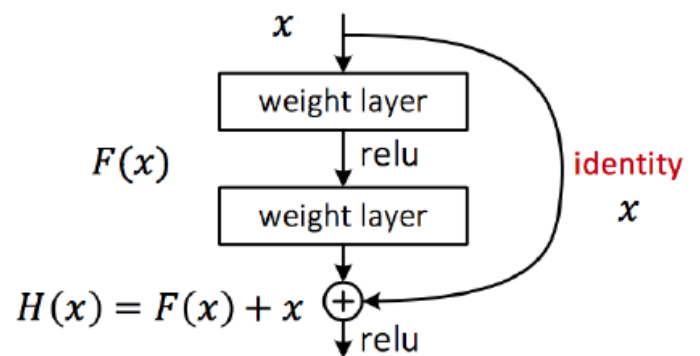
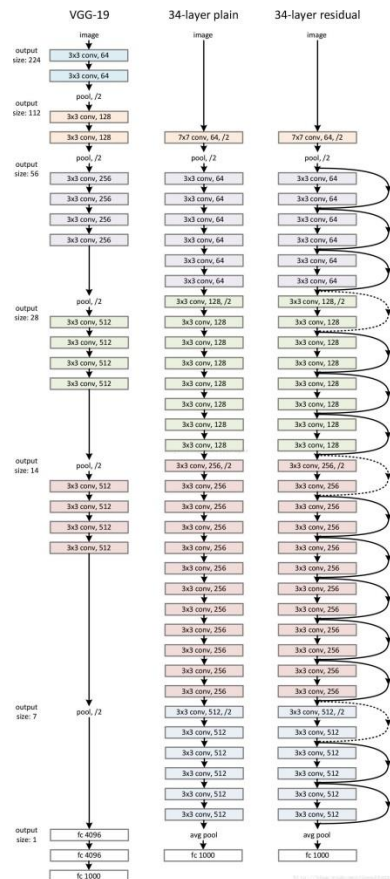
## 5. 深度殘差網路(Residual learning)

深度殘差網路的概念是為了解決當模型深度變深時，反而出現退化的情形(主因是**失誤率增加**，而非 overfitting)



如上圖，當神經網路的深度增加時，反而出現退化。

①深度殘差網路的形式



如上左圖，深度殘差網路可像最右邊的的神經網路形式，自行需要選擇哪幾層進行訓練。如此，當遇到較簡單的輸入，可以自動簡化模型(如左邊的神經網路)，減少失誤率。另外，在反向傳遞時，因為任兩層都有關係，淺層神經元的梯度也可直接傳遞到深層神經元，減少了**梯度消失**或**死亡神經元**帶來的影響(**平滑梯度傳遞**)。

上右圖則顯示了殘差神經網路的計算模式。對於每一層( $l$ 、 $l+1$ 、 $l+2$ 、 $\dots$ )，可以表示成：

$$x_{l+1} = x_l + F(x_l) \text{ (若不通過，輸出同為 } x_l \text{，不存在退化問題)}$$

$$x_{l+2} = x_{l+1} + F(x_{l+1}) = x_l + F(x_l) + F(x_{l+1})$$

最後可寫成：

$x_L = x_l + \sum_{i=1}^L F(x_i)$ ，此計算式將任兩層的關係表示成一個線性關係，對於梯度的計算更有利。

## 參考資料

1. 捲積(Convolution)

<https://zh.wikipedia.org/zh-tw/%E5%8D%B7%E7%A7%AF%E5%9B%BE%E8%A7%A3>

2. An Introduction to different Types of Convolutions in  
Deep Learning

<https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d>

3. Residual Networks (ResNet) - Deep Learning

<https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/>