國立雲林科技大學  
電子工程所

National Yunlin University of Science and Technology

Graduate School of Electronic Engineering

報告題目 :汽車駕駛者監控系統設計與實現

|  |  |
| --- | --- |
| 報告日期(Date)： | 2022/11/25 |
| 姓名(Name)： | 謝昕樺(碩一生) |
| 指指導教授(Professor)： | 蘇慶龍副教授 |
| E-mail： | [M11113020 yuntech.edu.tw](mailto:M10813007@yuntech.edu.tw) |
| 學號 (Student ID)： | M11113020 |

**本週進度(Progress of this week)**

* 高等數位訊號作業

11/11 Fri 13:00~19:00 (6.0 個小時)

11/11 Fri 21:00~03:00 (4.0 個小時)

11/12 Sat 16:00~00:00 (8.0 個小時)

11/13 Sun 19:00~00:00 (5.0 個小時)

* 機器學習運模作業

11/14 Mon 13:00~17:30 (4.5 個小時)

11/16 Wed 13:00~22:30 (9.5 個小時)

11/17 Thu 13:00~21:00 (8.0 個小時)

11/18 Fri 23:30~02:30 (3.0 個小時)

11/22 Tue 20:00~22:00 (2.0 個小時)

11/23 Wed 14:30~22:30 (8.0 個小時)

* 查看汽車駕駛監控系統相關論文

11/19 Sat 14:00~19:30 (5.5 個小時)

11/20 Sun 14:00~17:30 (3.5 個小時)

11/20 Sun 21:00~00:00 (3.0 個小時)

11/21 Mon 13:00~00:00 (11.0 個小時)

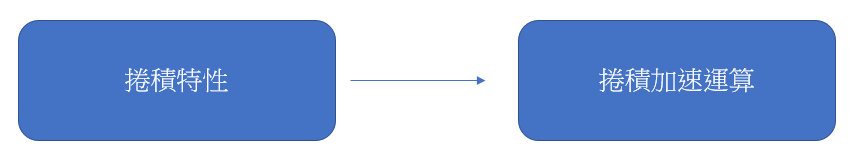
* 製作Meeting報告

11/24 Thu 16:30~18:00 (1.5 個小時)

11/24 Thu 19:30~21:00 (1.5 個小時)

共 84小時

|  |  |
| --- | --- |
| **11/14進度** | **11/25進度** |
| 1. 相機轉換視角演算法 2. 頭部姿勢固定演算法 3. 瞳孔區域偵測演算法 4. Opencl | 1. 捲積特性 2. 捲積加速運算 3. 高等數位訊號作業 4. 機器學習運模作業 |
| **待辦事項** | |

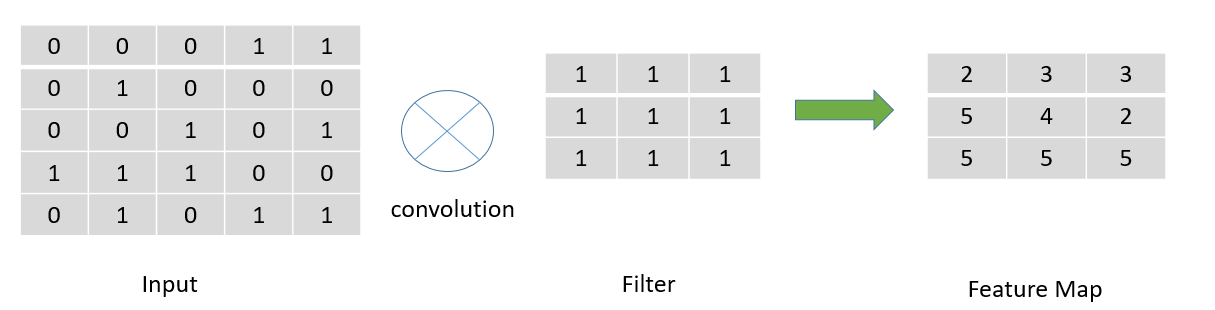


* 卷積運算

卷神經網路大多都是用於影像處理比較多，通過卷積以及

池化運算進行特徵提取，再進行卷積運算前要先定義濾波

器(Filter)的大小，通常是一個奇數M\*N的Martix(3 \* 3、5 \* 5)



上圖將每個Node做完Convolution後，會得出一張Feature Map，

接下來將Feature Map送入Activation functions內，得出來的數值

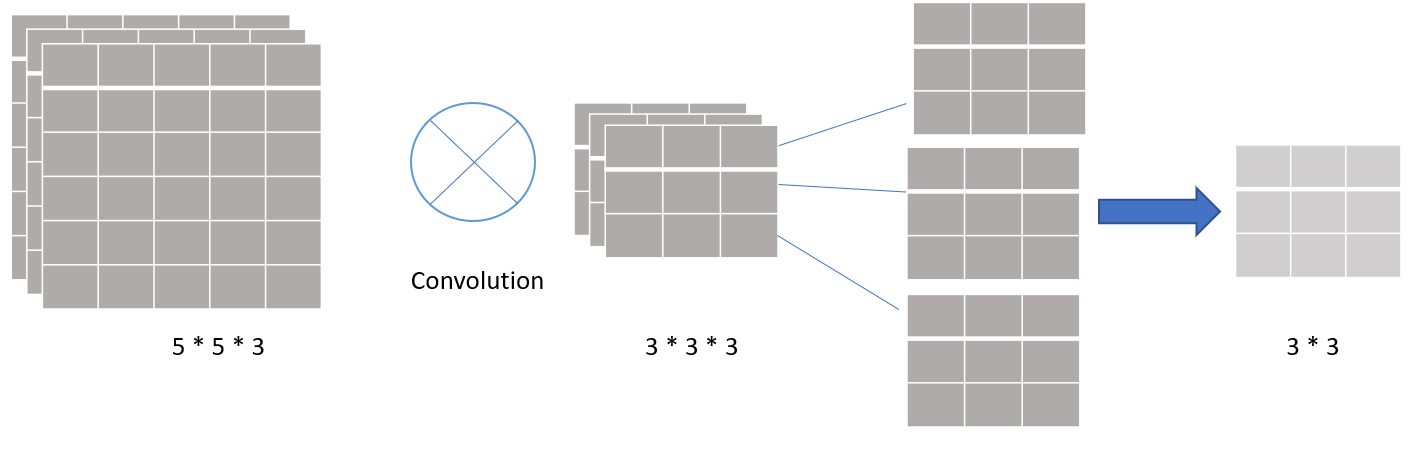
在進行Pooling

* 多層卷積層運算

若是輸入是一張黑白照片，其通道只有一層，但假設輸入一張彩色

照片，其通道就由R、G、B所組成，這樣通道就會由三層所組成，

這樣Filter也就需要三層

  
最後再將所有通道所對應的元素做相加會得到一張3\*3的Feature

Map

* Stride

當執行Convolution時，圖片是由左至右、上至下利用Filter進行運

算每次移動稱為步幅(Stride)，而計算Feature Map Size公式為

https://lh5.googleusercontent.com/pvwRjLkb2eZvbOBBnUbcRYzVOhDODuGGHsc7ydIq-2G2nQZWu6akSN0BEF8itT7vxHCP2xf3t0GsKKcO7SCrfWNHZvV-sabGIk1cV-Uki-euXnui5LOHyNmk3Bem3c58ectSxaLDPM71xqcx-cpLJbUFqvQ7RUfg0HrlQs21OjpnXDCiJ8WGS4khC8n3ZA

式(1-1)

以5 \* 5的Image為例，利用3\*3的Filter進行Conv，Stride為1，

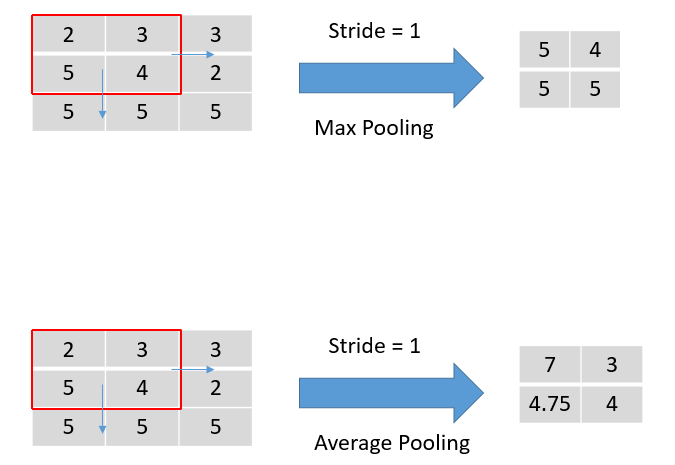
計算出來的Feature Map Size為3

* Pooling

對於人來說要分辨一樣東西，只要知道這項物品的特徵，而機器也

一樣，只需要擷取特徵點，不需要輸入整張圖片給機器做判斷，而

擷取特徵的方法分成Max Pooling和Average Pooling

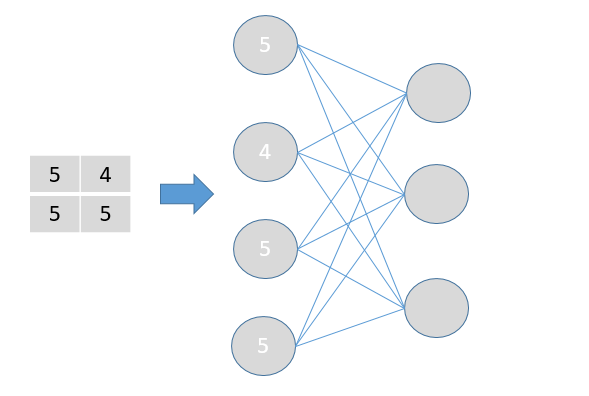
  
  
Max Pooling就是取紅框框內的最大值，由左至右由上至下取完整

張圖，而Average Pooling就是取紅框框內平均值，由左至右由上

至下做完整張圖

* Fully-connected layers

將Pooling過後的數值攤平成一個一維的Vector，接下來送入下一層Neural



* 卷積層和全連接層差異

全連接層原本是將所有Neural與Input進行連接，並且每一組的

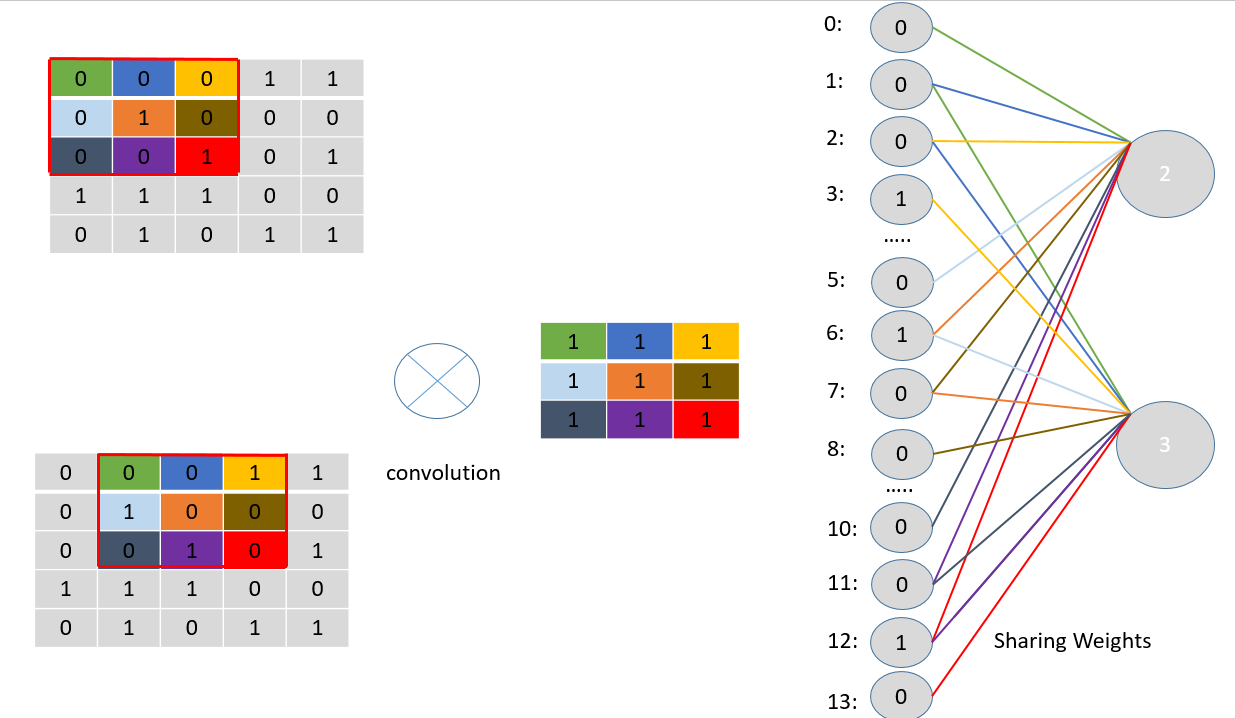
Weight都不相同，但卷積層與全連接層不一樣的地方在於局部連接

以及權重共享，以下圖為例，每個Neural所連接到的權重來自於

Filter的大小，原本用全連接層每個Neural都需要連接25個

Input，但是用卷積層只需要連接到9個Input，並且共用同一組

Weight



* Zero Padding

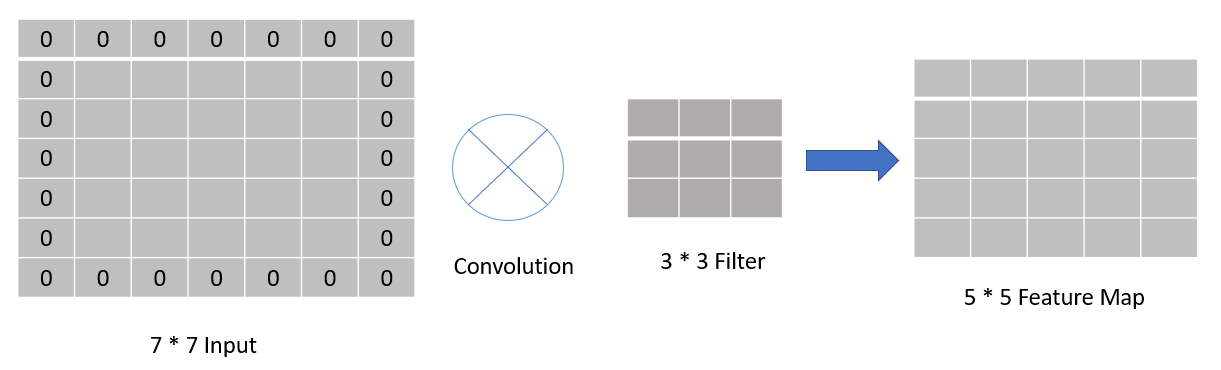
輸入圖像經由Conv以及Pooling後，圖像大小不僅會越變越小，邊

緣的特徵也會損失，若是在圖片輸入時就在其邊緣補0就可以解決

此問題，而此時Feature Map Size大小就會被改寫成式(1-2)

https://lh3.googleusercontent.com/F8j6m3ub4L-qyR-ahchjsU5ZjStbOcyAM3svwNRVwcBhJv4Rzw_kKP83wASAcyWoCVHqW-_UPEy6G56eFZNQPBRhjzkw5rW6Cogvjqmbu1HSV6kKoerVl4e1yfONT_R95zK44swjyS6Hph7RgWLAFugjaz9jXUt_dpMncCgT4_UkYhnebKDaP2ltdxzj6Q

式(1-2)



以5 \* 5的Image為例，利用3\*3的Filter進行Conv，Stride為1，

Padding為1，計算出來的Feature Map Size為5

* Feature Normalization

當輸入樣本特徵區間過大，分佈不均勻時會導致Loss Function難以

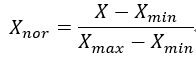
收斂，假設有兩組數值，A的範圍大概在10000，B的範圍落在0~1

之間，計算Gradient Descent時，B的影響就會變得非常小，而特

徵標準化分成兩種方法

* 正規化 :將數值縮放到0~1的範圍，如果資料範圍固定沒有極

端的最大值或最小值可以使用，公式為式(1-3)



式(1-3)

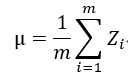
* 標準化: 將資料轉成平均值為0，標準差為1，若資料多為極端

值可以使用，公式為式(1-4)， = 所有總數相加取平均

(平均差)， =( 每組數值 –) 相加取平均開根號

https://lh5.googleusercontent.com/7ubb40WYcnDVa2Z6i4oxIAfKpY4fTbhhpoh8o7GBylYxVq_KP_-gUdgxOERreJ8lWWZGFbZRENOigxUl4XMxLgOI1Hklx7rcSgdKZzz0djDFVktTcuqQuVMPyY9wJamG1yk8j7XQpZwcNkpGWiq8Ac_61Qrkl-oUOPK9rtChT_f7uYFZOJIn3nkMo64_hQ

          式(1-4)



式(1-5)

https://lh3.googleusercontent.com/g4wgIgDIC0nCBnIfvduC55yvw8YaqzmMqFa5DztlSQuZb4I7Mnjc0XIPzaTfw_hijjx2ZhCelNtQQY0HctCNBZFuvUxZI344qn38n5aeRPt7mv1nQZTbEfRO29VMlAz3a22B9rt0We1hwV0v3kdS3pPzOoESGlsd0fWYmpnwx-HRjZxLUj8GlhbCyL5PrQ

式(1-6)

* Batch Normalization

雖然在輸入的時已經有將Feature Normalization，但是經過每組不

同的Filter後，有可能讓Nerual內的資料差距拉大，當這些資料進

入Activation Function後，會因為輸入資料太大或太小導致資料分

布不均勻，以Sigmoid為例，差距太大或太小的資料會分布在0~1

之間，這樣中間數值沒有被參考到，因此就需要數值在進入

Activation  Function時，進行Normalization的動作，而使用Batch

Normalization的優點為

* 加速Neural的訓練可以更快收斂

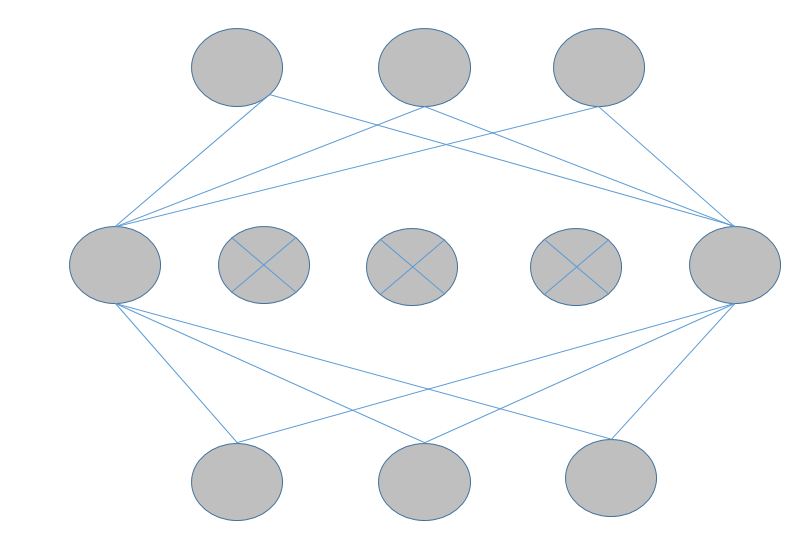
* 優化器可以有更大的Learning Rate，並且讓初始權重

更簡單

* 緩解梯度消失神經元也能有更多Activation Function
* Dropout

Dropout是一種對抗過擬合的正規化方法，其原理是隨機丟棄下層

Neural，讓模型不要過於依賴某些資料



以上圖為例，有5個Nerual裡面有3個被丟棄了，因此可以得出

Dropout機率為60%，被丟棄的Nerual不會Fordward pass任何訊

息，在Backpropagation時，被丟棄Neural的Weight為0，因此不

會過度依賴某些Neural

* Xnor-Net

卷積網路主要由卷積層所組成，加快卷積層運算對整個網路性能至關

重要，目前有兩種方法可以量化網路，分別為Binary-Weight-Network

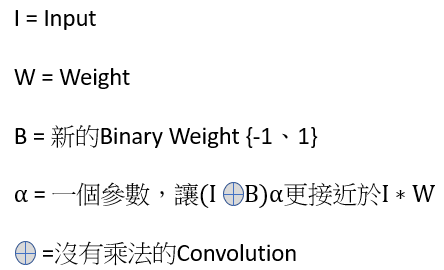
(只有權重被量化) 和 Xnor-Network(權重和 Input 量化)，Binary-Weight-N

etwork是將Weight的值轉換成{}，與Input值進行運算，而Xnor-Ne

twork是將權重和Input轉換成{}，利用運算



式(1-7)



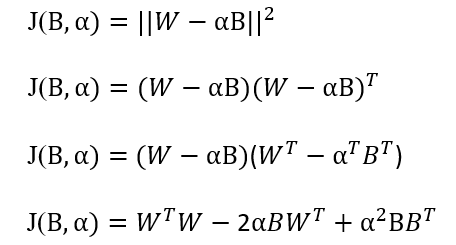
由於兩邊的Input都一樣，因此可以將其改寫成式(1-8)

式(1-8)

J( 式(1-9)

式(1-10)

將式(1-9)展開為式(1-11)



式(1-11)

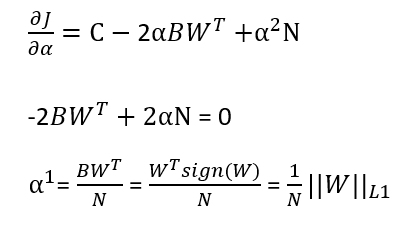
將用C表示，用N表示可以將式改寫為式(1-12)



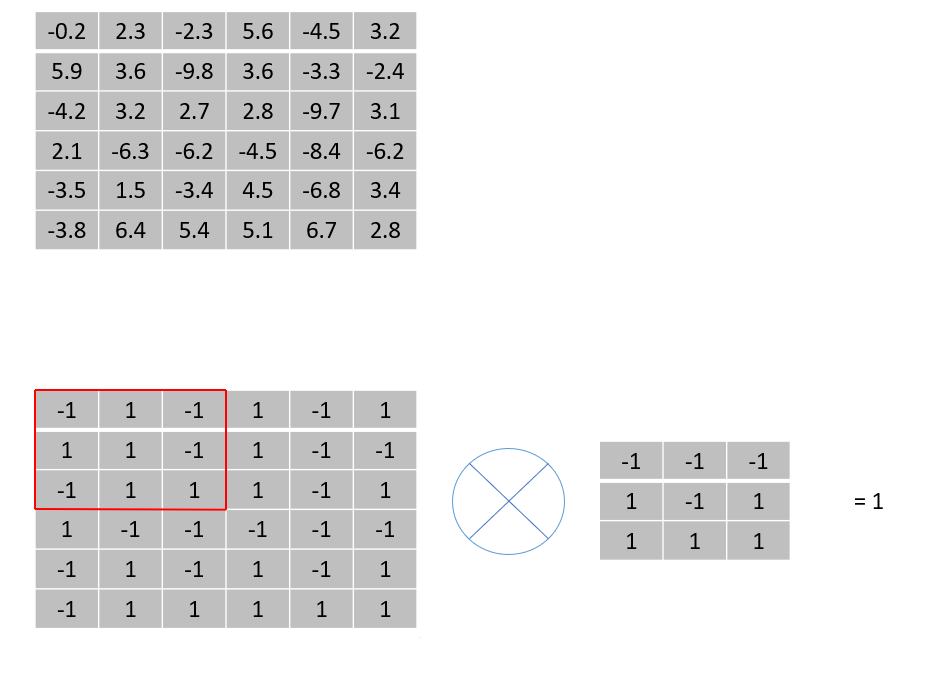
式(1-12)

接下來要求最小值

式(1-13)



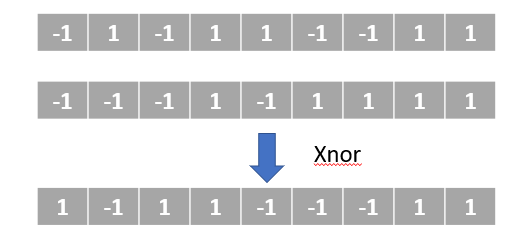
式(1-14)



首先將Input做歸一化動作得出C，若該值小於C設為0，大於設為1，

這邊為了方便計算，將小於0數值設為，大於0數值設為 1以傳統

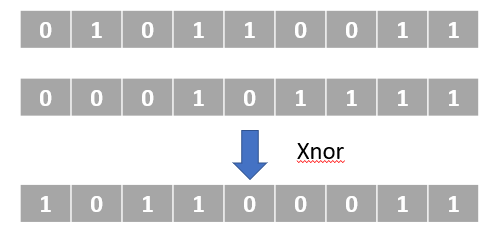
的Conv去計算紅框框數值得出結果為1



接下來將紅框框與 Filter攤平做Xnor計算，計算出來的結果與傳統

Conv結果一樣

在實際運算時，需要將值設為0，所以將圖改成



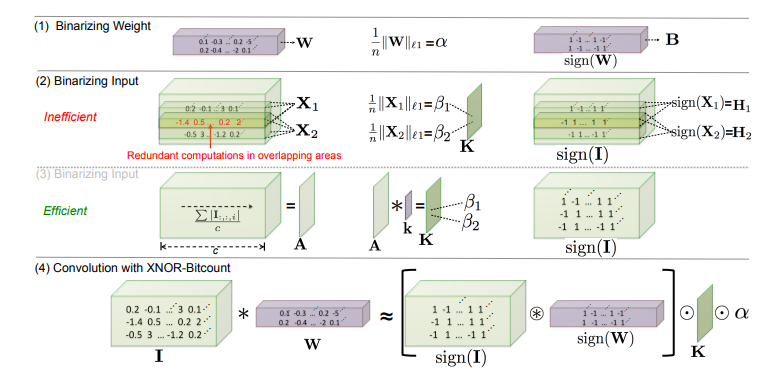
接下來透過式(1-15)將數值做轉換，轉換過後的值也跟傳統Conv數值

一樣

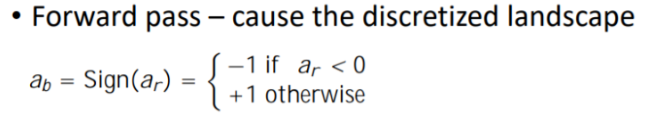
式(1-15)

Onecount = Filter內數值為1的總數

Filter\_\_Total = Filter內部總共幾組數值

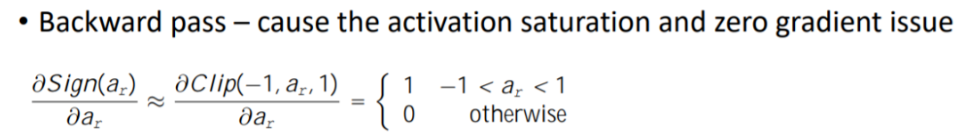
Conv = 2 \* 5 - 9 = 1

Xnor-Net的Forward pass是由Sign(W)={}所組成

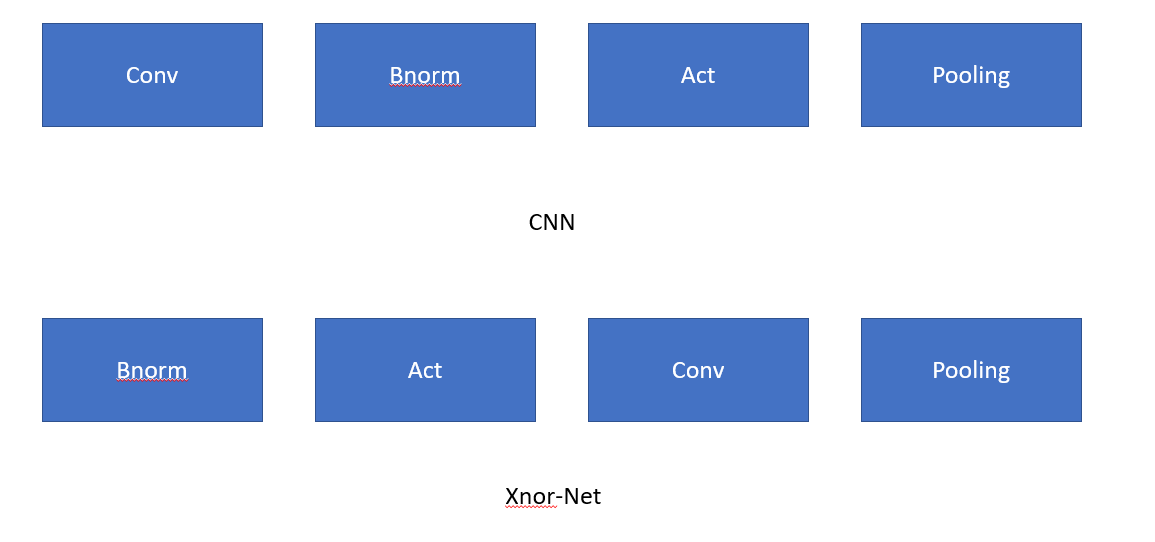


若是經由Backpropagation後假設數值超出了{}的範圍，對應的導

數會歸 0



所以Xnor-Net的捲積層結構和一般CNN結構不一樣



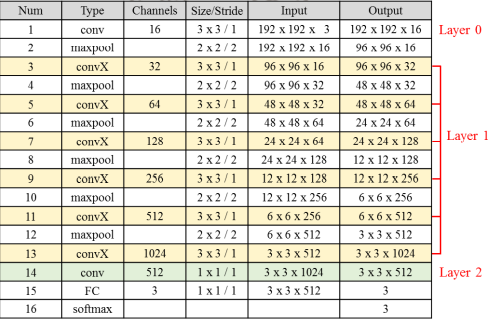
假設上圖的Activation Function為Relu，以CNN架構去執行Xnor-Net時，

會讓Pooling的值都是0或者正整數，這樣會讓下一層的特徵不好提取，

但是用Xnor先在Conv之前執行Activation Function時，可以讓數值非0及

1，這樣方便我們進行Xnor-Net計算，經過Pooling時也能有更多特徵提

取



上圖為學長本次論文架構圖，第一層輸入和最後一層不做Xnor-Net運

算，原因為若做了Xnor優化會大幅降低準確率， 而為了降低運算量，

每一層的Pooling以Stride為2的方式減少，同時又想要提取特徵因此增

加通道數的維度，最後再用一層1\*1的Conv來降維，以Softmax分成三

類