Embedded Vision Intelligent Laboratory

嵌入式智慧影像分析與實境界面
Fall 2021

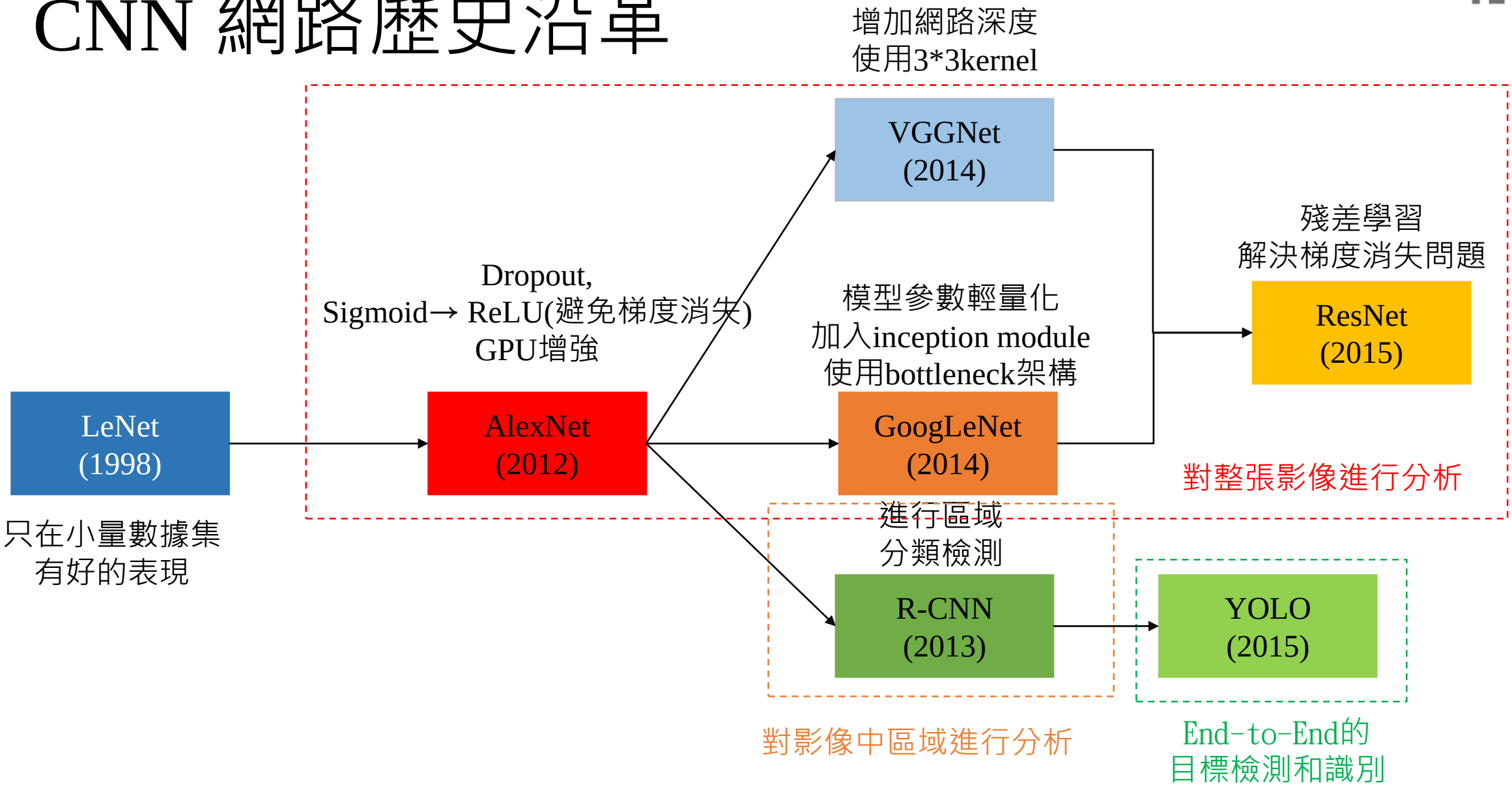
Instructor：Yen-Lin Chen(陳彥霖), Ph.D.
Professor
Dept. Computer Science and Information Engineering
National Taipei University of Technology

Lecture 8

CNN神經網路

CNN 神經網路發展

CNN 網路歷史沿革


增加網路深度
使用3*3kernel
VGGNet
(2014)
殘差學習
解決梯度消失問題
ResNet
(2015)
Dropout,
Sigmoid→ ReLU(避免梯度消失)
GPU增強
模型參數輕量化
加入inception module
使用bottleneck架構
LeNet
(1998)
AlexNet
(2012)
GoogLeNet
(2014)
對整張影像進行分析
只在小量數據集
有好的表現
進行區域
分類檢測
R-CNN
(2013)
YOLO
(2015)
對影像中區域進行分析
End-to-End的
目標檢測和識別

VGGNet

VGGNet 介紹

- VGGNet在2014年 ILSVRC 的分類比賽中拿到了第二名
- VGGNet改良了AlexNet，將網路結構加深，進而得到更好的結果。

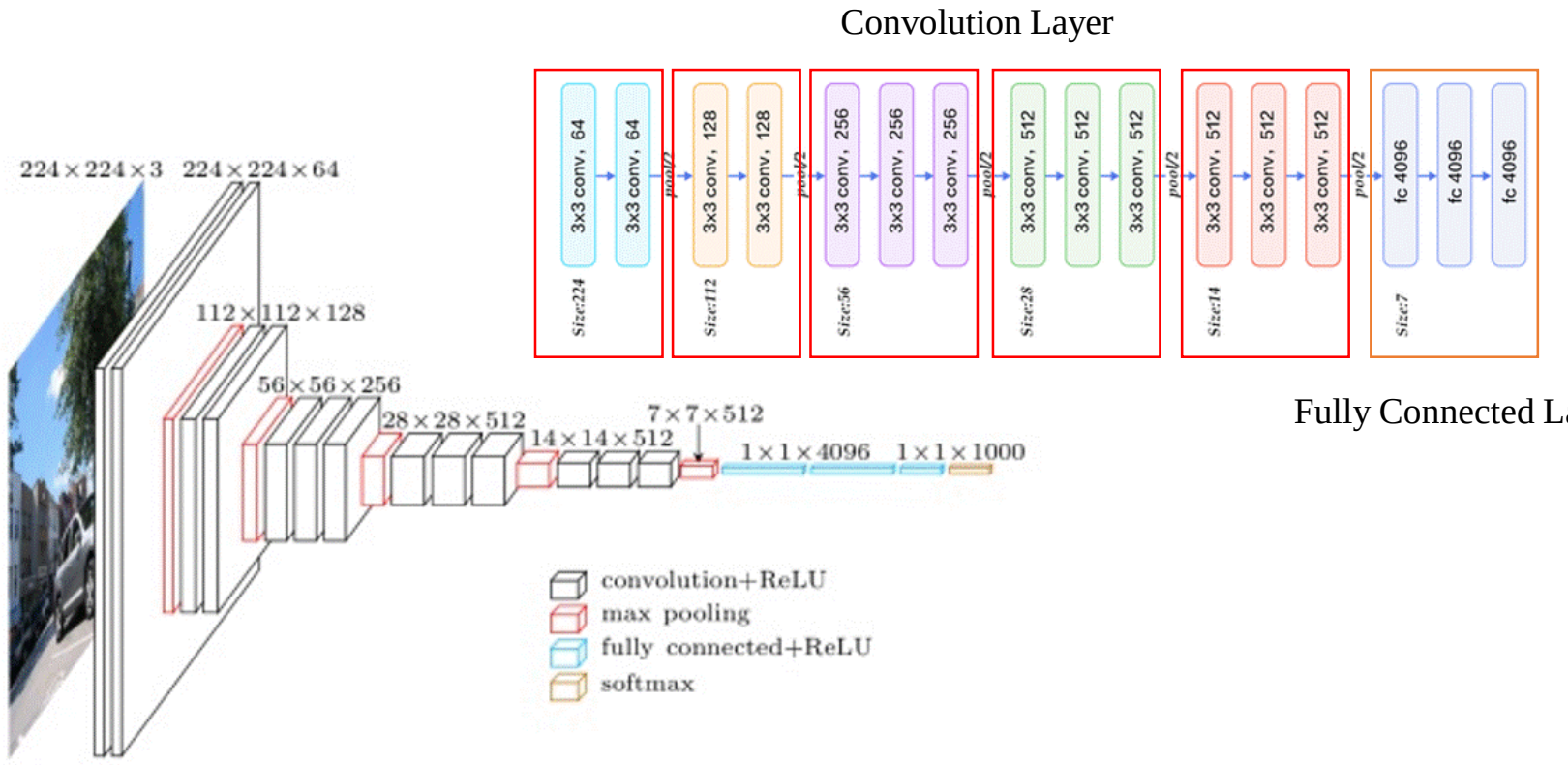
VGGNet 特點

- 架構全部使用3x3的kernel當作捲積神經層和2x2的kernel當作池化層，架構簡單統一
- 證明較深的層數能提高效能
- 通過多個3x3不斷地加深網路，也增加了許多參數，運算速度較其他框架慢。

VGGNet 架構

- Input image size: 224*224*3
- 每一個卷積層皆使用3*3的kernel，根據不同深度決定卷積層層數
- 池化層使用2*2的kernel進行Maxpooling
- 全連接層，加上ReLU，4096個神經元
- 右下圖為VGG16架構

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					



GoogLeNet

GoogLeNet 介紹

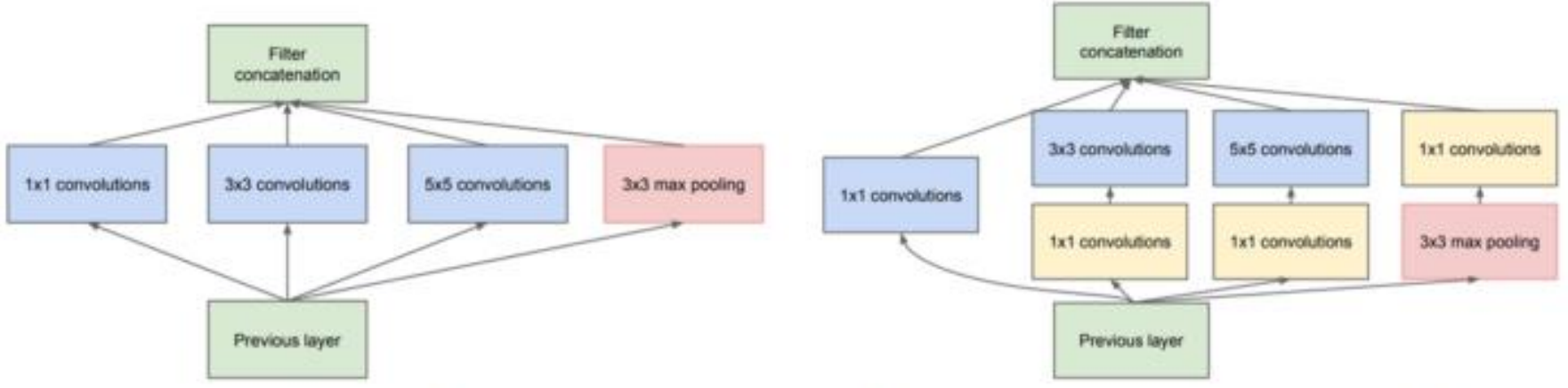
- GoogLeNet 最早是發表在 Google 的 Paper：Going deeper with convolutions，裡面介紹了 Inception V1/GoogLenet 架構，並在 ILSVRC-2014 比賽中在分辨項目第一名(Top-5 Error=6.67%)
- GoogLeNet只有約 680萬個參數，比 AlexNet 少九倍，更比 VGG-16 少二十倍，也就是說 GoogleNet 的模型更為輕巧。

GoogLeNet 特點

- 透過多個Inception module串聯成大網路。
- 使用Bottleneck架構，在降低深度，但不降低原輸入二維的維度情況下，降低計算量。
- 使用Auxiliary Classifiers解決梯度消失問題

Inception module 架構

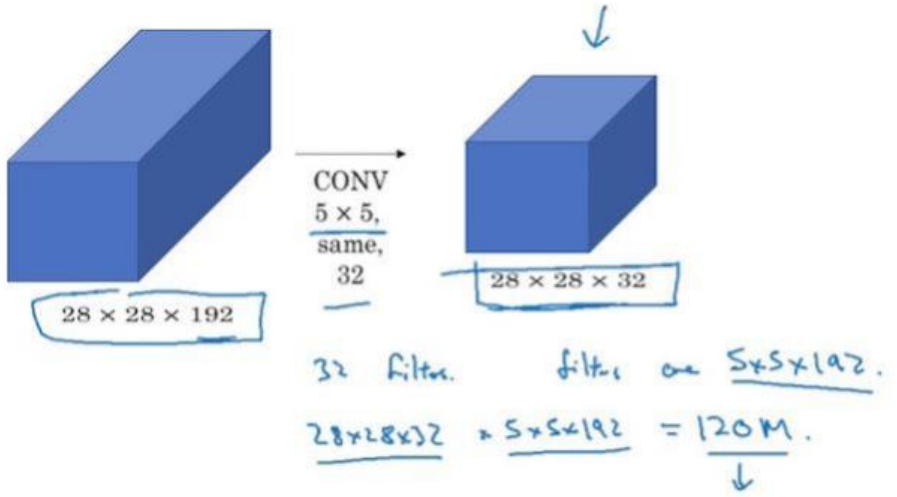
- Inception module 一開始採用下圖 (a)，透過三種不同大小的卷積以及 3x3 Maxpooling 萃取出不同的特徵，再將這四個結果以通道軸串接在一起。這樣增加網路寬度的方式，能夠擷取更多圖片的特徵與細節。
- 但若是這四個結果的尺寸不同，則卷積層與池化層皆使用 padding=”same”、stride=1，以確保輸入特徵圖的尺寸
- 後來為了降低訓練參數量，Inception module 在原先的架構中加入一些 1x1 卷積層 (圖b)。改良後的 Inception module 會先透過 1x1 卷積層降低輸出通道數後，再接上原本的卷積層。此外，由於池化層沒辦法降低通道為數，因此輸入特徵圖在通過 3x3 Maxpooling 後，會再使用 1x1 卷積層來降低輸出通道數。

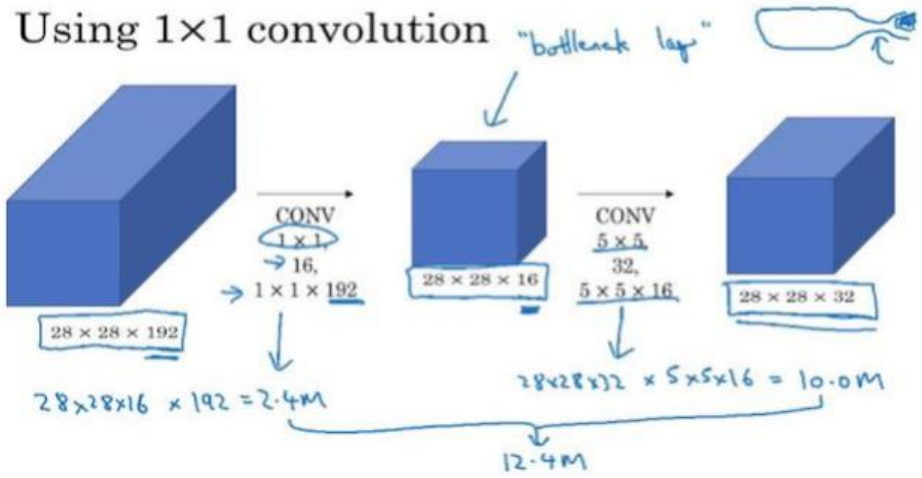


(a) Inception module, naïve version　　(b) Inception module with dimension reductions

Bottleneck 架構

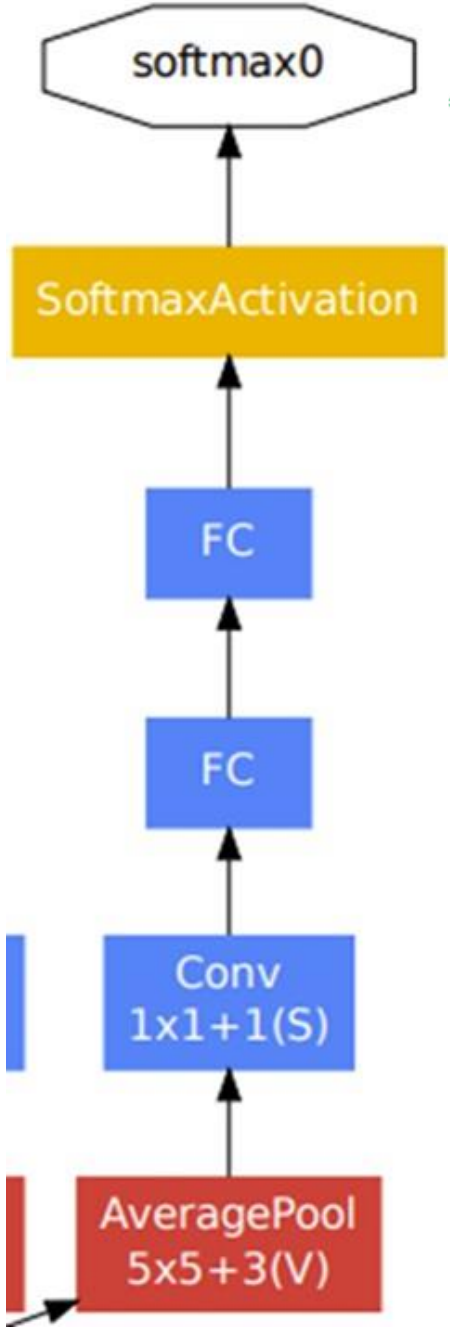
- 在一般 convolution layer，在輸入維度相當高的情況下，做 convolution 其計算量是相當大的，尤其在希望不遺失局部的細節，仍使用較小的長度的 filter 的情況下。
- 如左上圖，在 28x28x192 的輸入中使用 5x5 的 filter，會達到約120M的運算量($28 \times 28 \times 32 \times 5 \times 5 \times 192$)。
- 但若透過 1x1 convolution layer 引進 bottleneck 結構而達到降維的效果後，再執行 5x5 convolution operation，將會有效地減少所需的計算數目。
- 可見右下圖，透過 1x1 convolution layer 減少了原輸入的 channel 大小，從 192 到 16，再執行 5x5 convolution operation 可發現總共的計算次數約12.4 M($28 \times 28 \times 16 \times 192 + 28 \times 28 \times 32 \times 5 \times 5 \times 16$)，比起在原輸入直接執行 5x5 convolution 減少了將近 10 倍。





Auxiliary Classifiers 架構

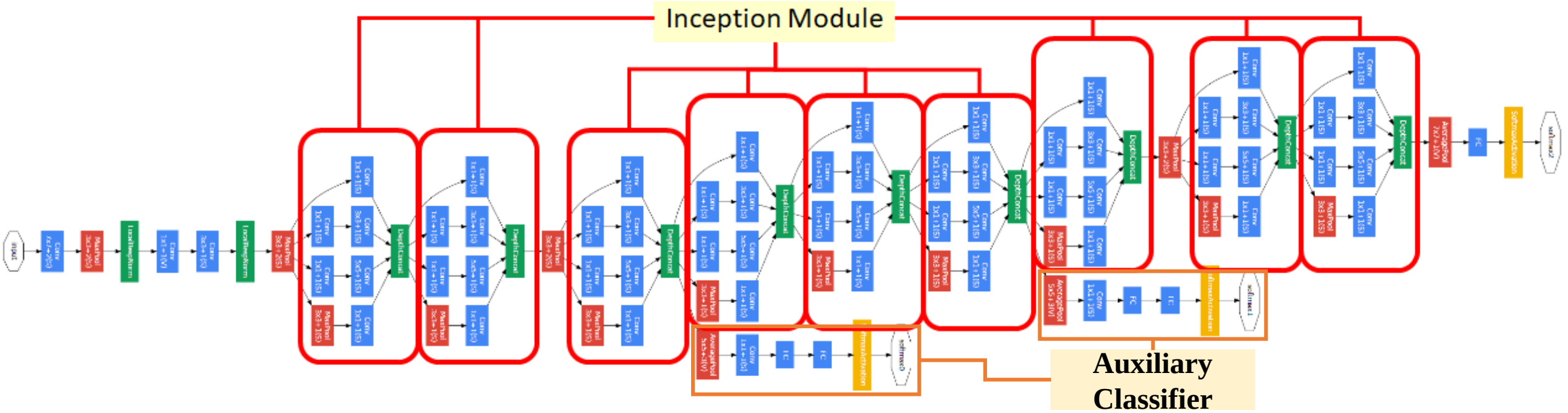
- 在兩個不同層的 Inception module 輸出結果並計算 loss，最後在將這兩個 loss 跟真實 loss 加權總和，計算出總 loss，其中 Inception module 的 loss 權重值為 0.3
- 公式:
  - total_loss = real_loss + 0.3 * aux_loss_1 + 0.3 * aux_loss_2



GoogLeNet 架構

- 使用9個Inception Module組成
- 使用分類輔助器(Auxiliary Classifiers)避免梯度消失問題

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								



參考資料

- Day02 VGGNet（CNN延伸應用）(coderbridge.com)
- Neutrino's Blog: GoogLeNet 簡介與小實驗 (tigercosmos.xyz)
- 卷積神經網絡 CNN 經典模型 — GoogleLeNet、ResNet、DenseNet with Pytorch code | by 李謦伊 | 謦伊的閱讀筆記 | Medium
- 10. 深度學習甜點系列：全面啟動 - iT 邦幫忙::一起幫忙解決難題，拯救 IT 人的一天 (ithome.com.tw)