# Day 20

# 深度學習與電腦視覺

## 學習馬拉松

cupoy 陪跑專家：楊哲寧

CUPOY

# 深度學習理論與實作

## Classic CNN Backbone
## (經典CNN框架)

# 重要知識點

CUPOY

- 了解ResNet中殘差網路的概念
- 了解如何導入殘差網路於Inception Block 中
- 本次內容與實作會拆成兩個parts，請參考Part1、Part2

CUPOY

# Part01

# ResNet

CUPOY

ResNet為2015年ImageNet classificatoin的冠軍，其網路層數打破過往紀錄。

| Model | AlexNet | Vgg | InceptionV1 | ResNet |
|---|---|---|---|---|
| 推出年 | 2012 | 2014 | 2014 | 2015 |
| 層數 | 8 | 19 | 22 | 152 |
| 卷積層數 | 5 | 16 | 21 | 151 |
| 卷積核大小 | 11,5,3 | 3 | 7,1,3,5 | 7,1,3,5 |
| Inception(NIN) | - | - | + | - |
| 全連接層 | 3 | 3 | 1 | 1 |
| 全連接層大小 | 4096*2,1000 | 4096*2,1000 | 1000 | 1000 |
| Top-5 error | 15.3% | 7.3% | 6.7% | 3.57% |

參考來源： IT Read01

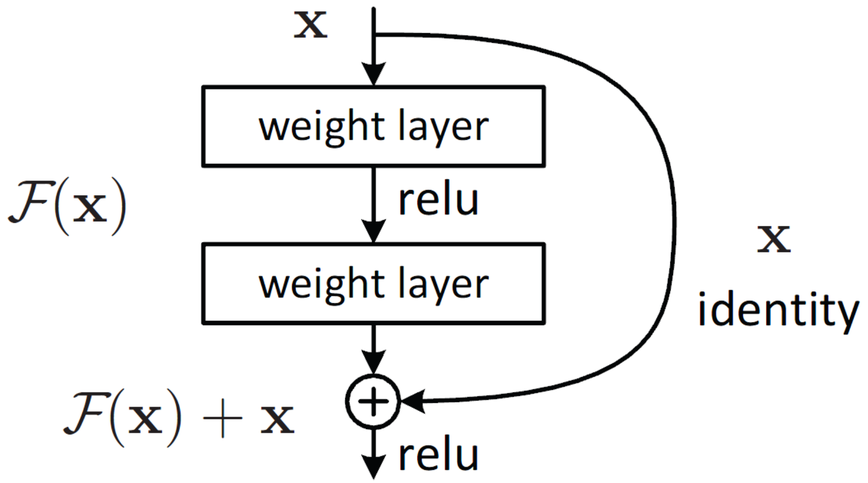
# ResNet

CUPOY

## ResNet 重點

# ResNetV1-Residual Block (Identity Block)

- 在ResNet問世以前，神經網路結構都不會太深，主要是由於梯度消失(vanishing gradient)的問題。
- 下圖為經典的殘差結構，將輸入的input與經過2–3層的F(x)跨接並相加，使輸出表示為**y=F(x)+x**，這樣的好處在於反向傳播時能保證至少會有一個1存在，**降低梯度消失** (vanishing gradient) 發生的可能性。



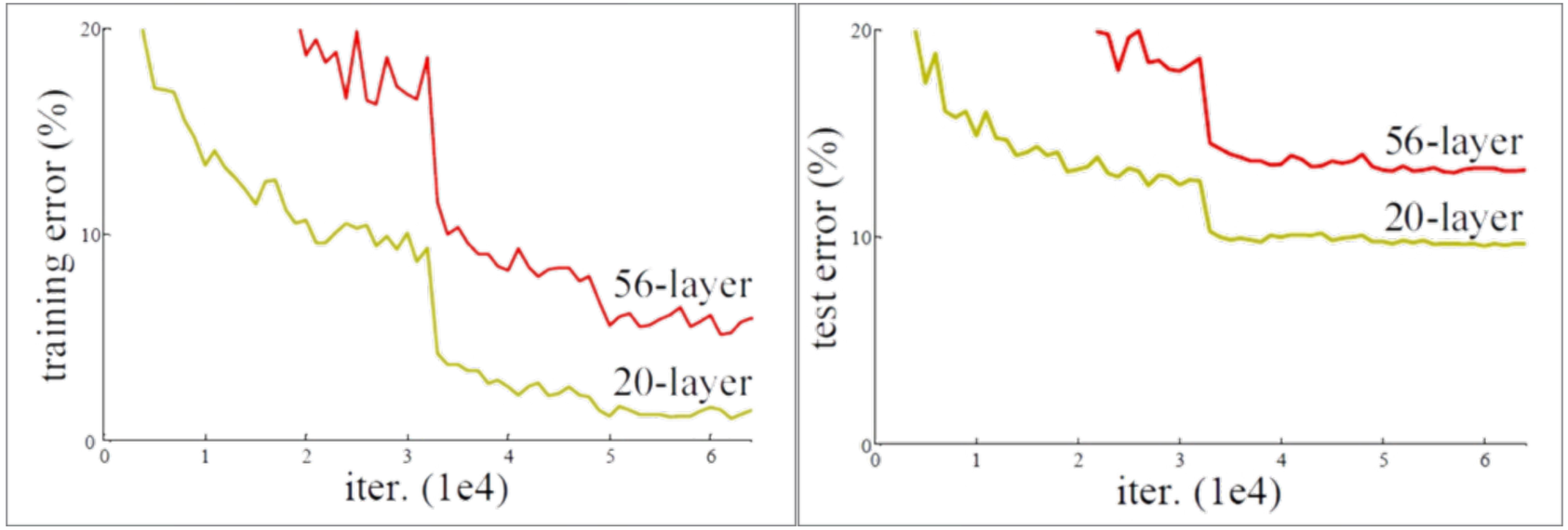

**Residual Block**

參考來源： Cornell University

# InceptionV4, Inception-Resnet

CUPOY

沒有殘差網路的幫助下，模型結構越深反而造成準度下降。




參考來源： Arxiv

# ResNet-Residual Block

CUPOY

我們在做Back Propagation時，如果要求x的梯度，我們必須利用輸出**y對x偏微分**，當網路很深時容易造成**梯度消失**，但如果鍊鎖率中每一個環節都**保有一項x**，自己對自己微分後都保有一個**1**，梯度消失的可能性就大幅降低，因此可以搭建更深的網路。

$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\frac{\partial \mathbf{x}_L}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\left(1 + \frac{\partial}{\partial \mathbf{x}_l}\sum_{i=l}^{L-1}\mathcal{F}(\mathbf{x}_i, \mathcal{W}_i)\right)$$

**每一層都保有一個1**

參考來源： Arxiv

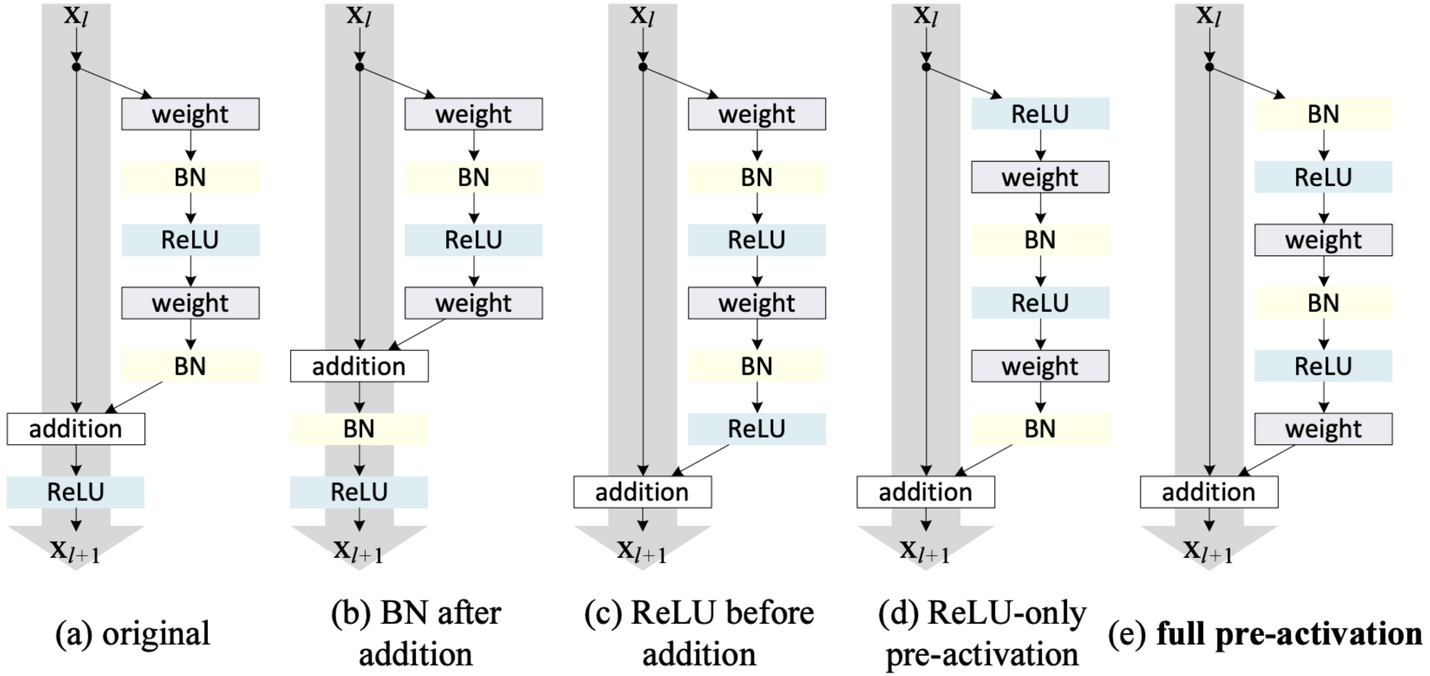
# ResNet-Residual Block

- 實現其實相當簡單，我們只需要將Block輸出**加上輸入的值**即可。
- 要注意的是輸入與輸出**Feature Map**必須有相同**大小跟深度**，不然無法直接相加。

```
def Residual_block(input_tensor, kernel_size, filters, stage, block):
    filters1, filters2, filters3 = filters
    conv_name_base = 'res' + str(stage) + block + '_branch'
    bn_name_base = 'bn' + str(stage) + block + '_branch'

    x = Conv2D(filters1, (1, 1), name=conv_name_base + '2a')(input_tensor)
    x = BatchNormalization(axis=3, name=bn_name_base + '2a')(x)
    x = Activation('relu')(x)

    x = Conv2D(filters2, kernel_size,
               padding='same', name=conv_name_base + '2b')(x)
    x = BatchNormalization(axis=3, name=bn_name_base + '2b')(x)
    x = Activation('relu')(x)

    x = Conv2D(filters3, (1, 1), name=conv_name_base + '2c')(x)
    x = BatchNormalization(axis=3, name=bn_name_base + '2c')(x)

    x = layers.add([x, input_tensor])
    x = Activation('relu')(x)
    return x
```

# ResNetV2-Residual Block

ResNetV2同樣由Kaiming He團隊提出，承襲ResnetV1的殘差概念，但在**Identity branch(左線)與 Residual branch(右線)**上做了一些更改。

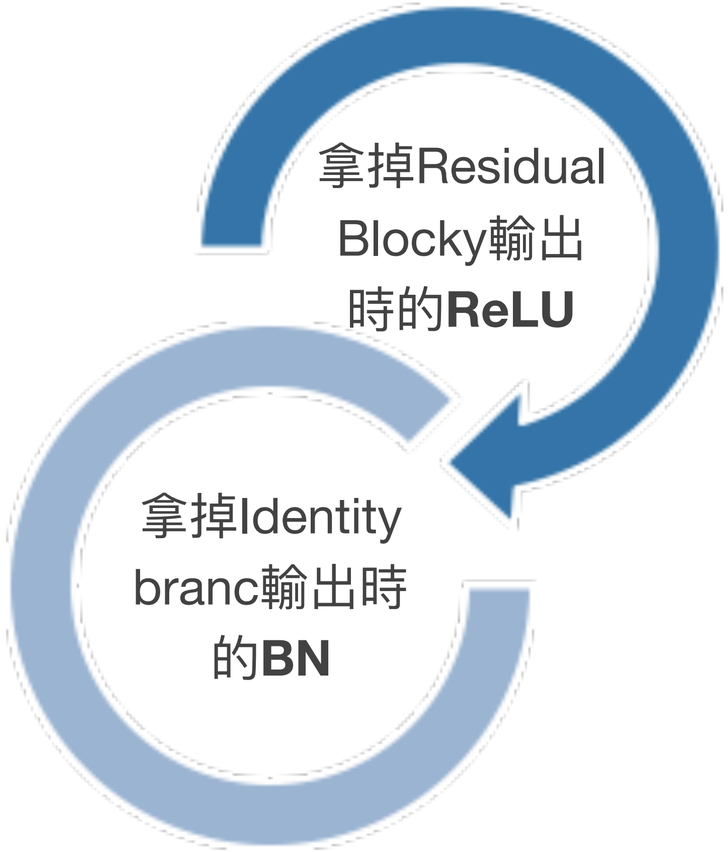



多種 Residual Block 的設定

參考來源： Arxiv

# ResNetV2-Residual Block

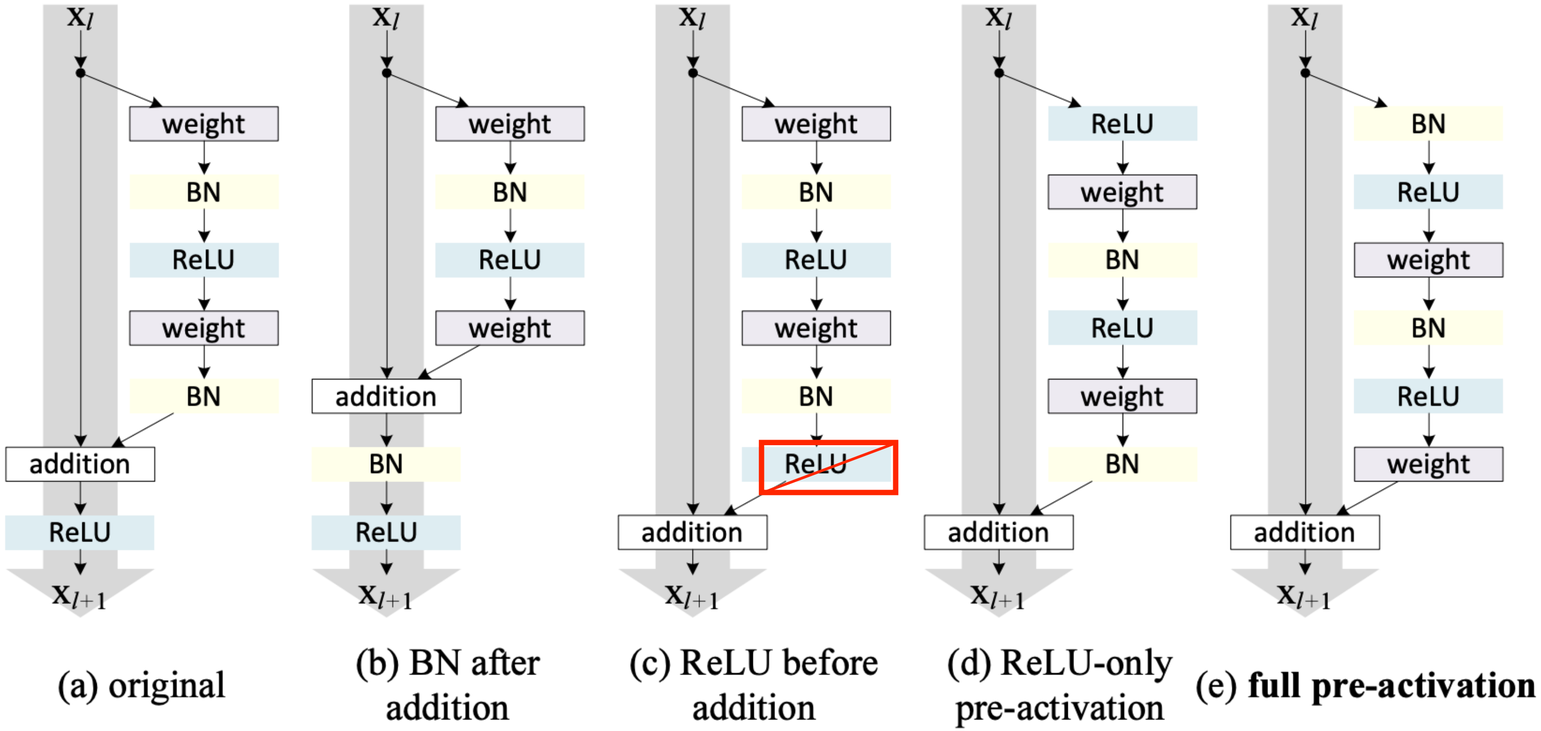
CUPOY



# ResNetV2-Residual Block

- 拿掉Residual Block最後一層的ReLU

  作者認為，ReLU接在每個Residual block後面會導致Forward propagation陷入單調遞增，降低表達能力。



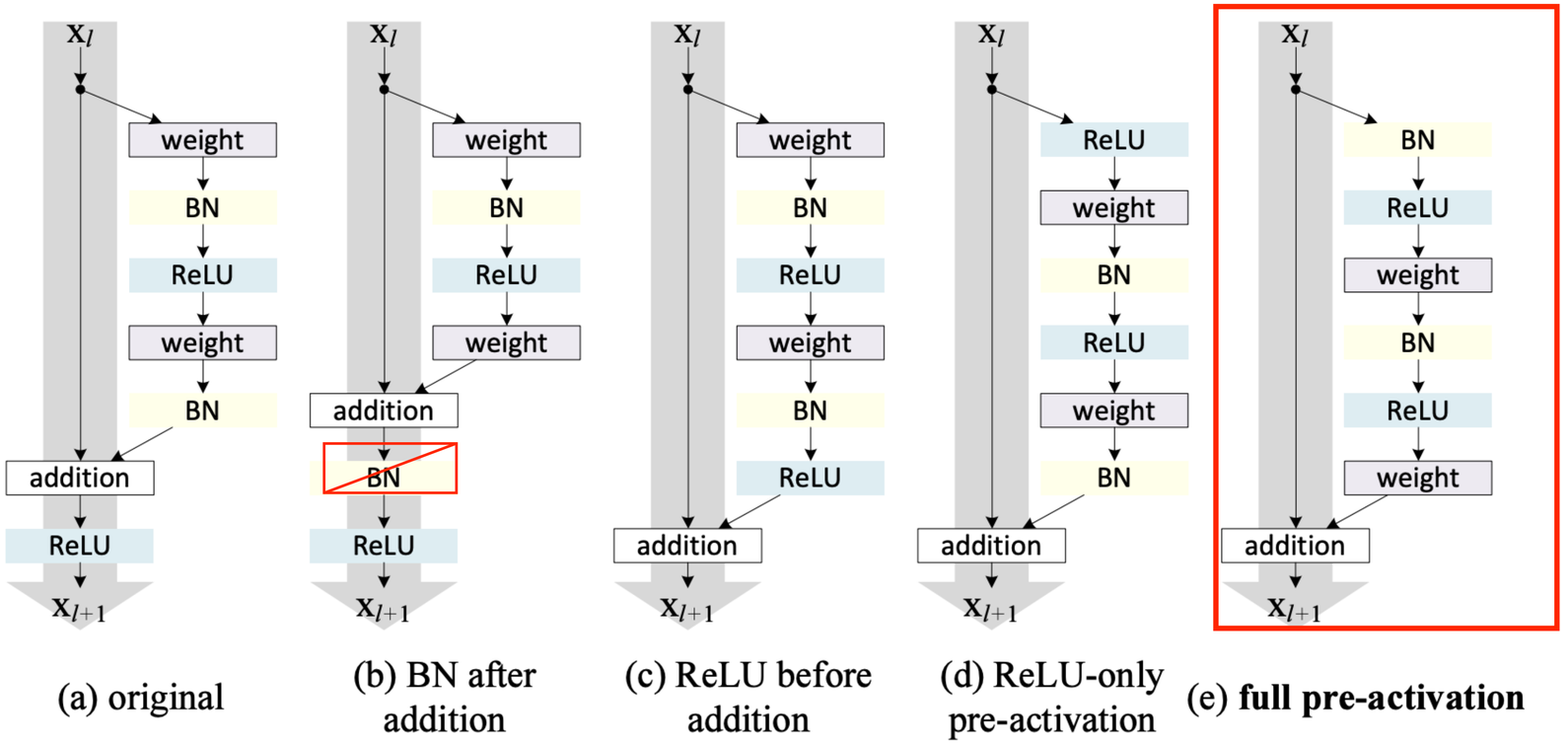

多種Residual Block的設計

參考來源： Arxiv

# ResNetV2-Residual Block

## 拿掉Identity branch後的 **BN**

BN層會改變**Identity branch的訊息分佈**，造成收斂速度下降。論文中也用到Inception Block中，1*1 kernel壓縮深度的技巧，最後再用1*1 kernel回放深度，藉此降低運算。




最終設計

多種Residual Block的設計

參考來源： Arxiv

CUPOY

# Part02

# InceptionV4, Inception-Resnet

CUPOY

- InceptionV4 重點

  設計多種不同的 **Inception Block**

- Inception-Resnet 重點

  結合 **殘差結構** 於 **Inception Block** 中，可以顯著加速 Inception 的訓練。

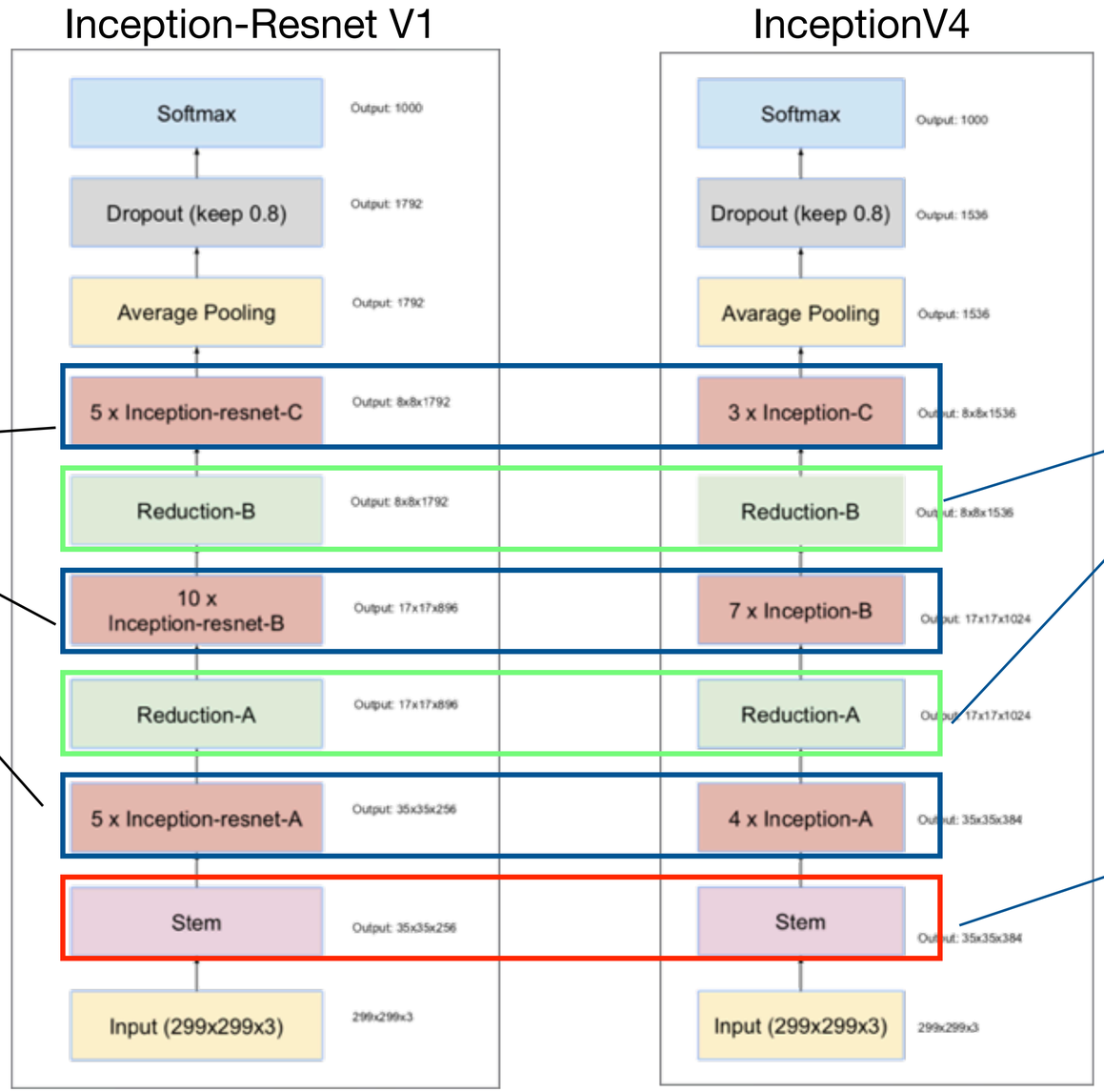
# InceptionV4, Inception-Resnet

CUPOY

Inception-Resnet像是加入殘差結構的V3、V4版本：

Inception-ResNet v1 的運算量和 Inception v3 的接近。
Inception-ResNetv2 的運算量和 Inception v4 的接近。

# InceptionV4, Inception-Resnet 結構




不同設計的Inception Block, -Resnet版本又加入了**殘差結構**

主要用來降低 Feature Map 尺寸

提取淺層語意訊息

參考來源： Arxiv

# InceptionV4, Inception-Resnet Stem結構

Inception_Resnet V1

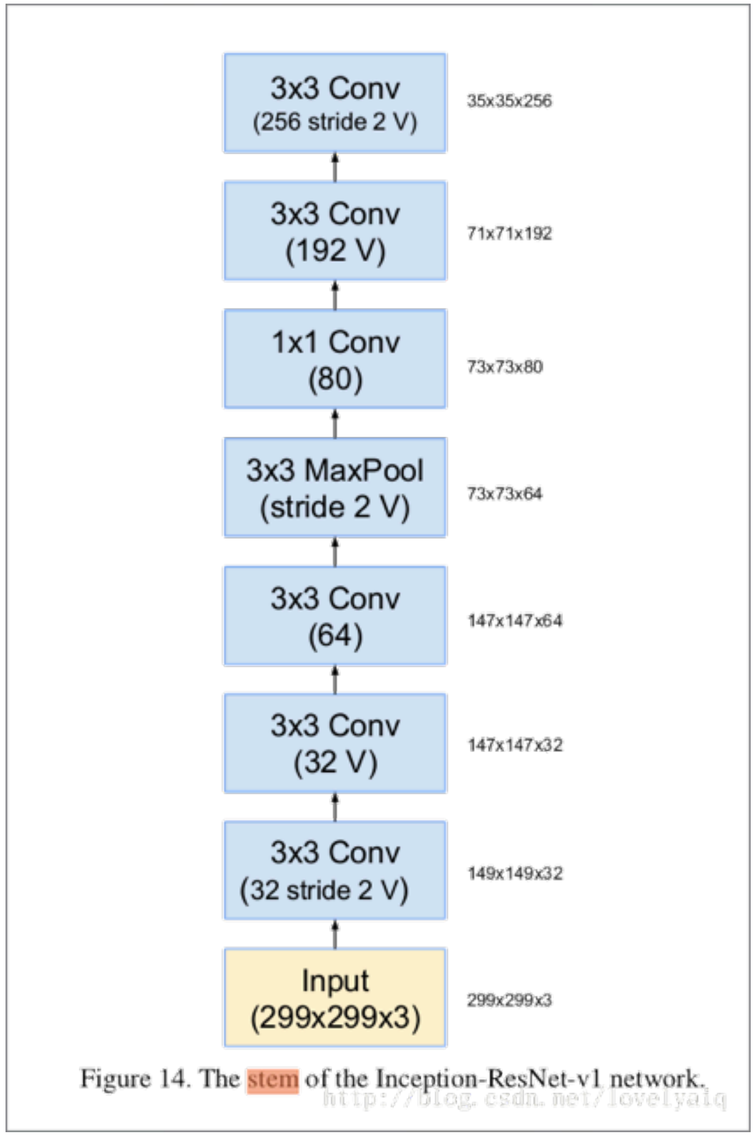
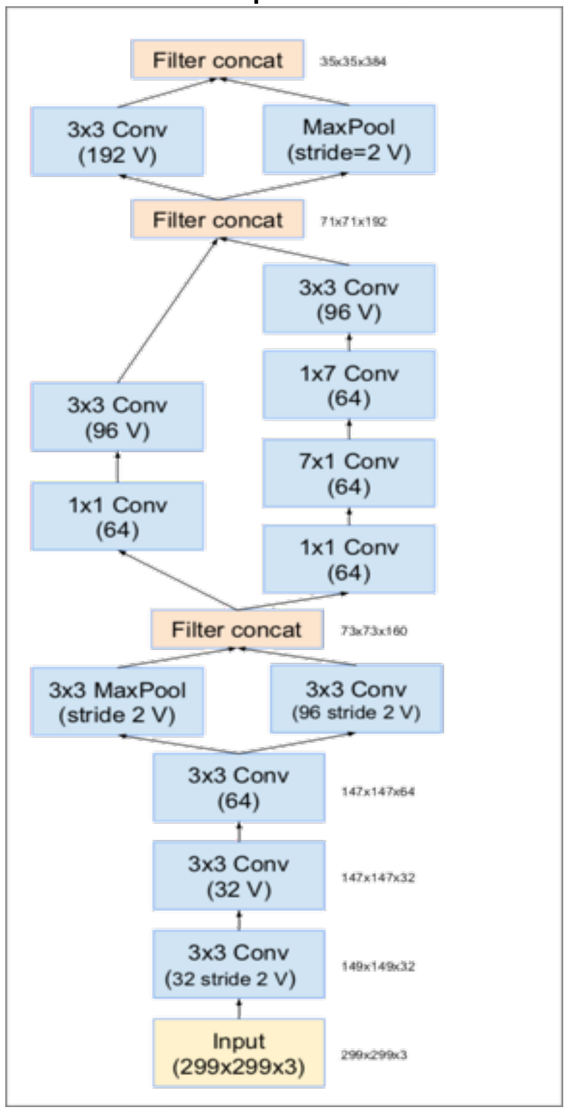



Figure 14. The stem of the Inception-ResNet-v1 network.



InceptionV4的Stem結構更為複雜，用來提取更豐富的淺層語意訊息。

InceptionV4

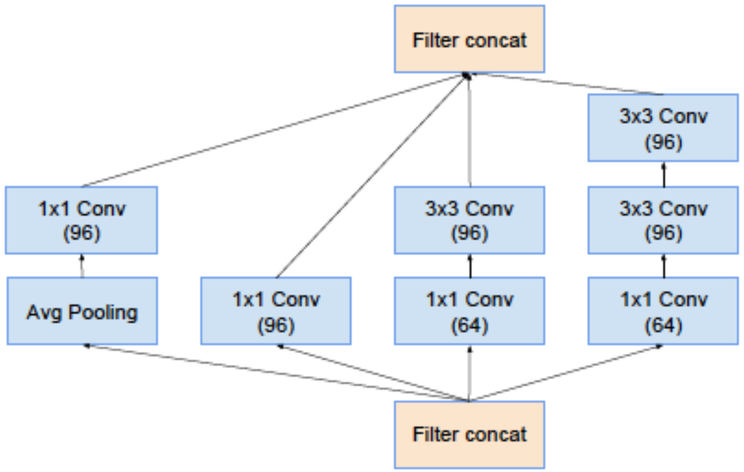



參考來源： Arxiv

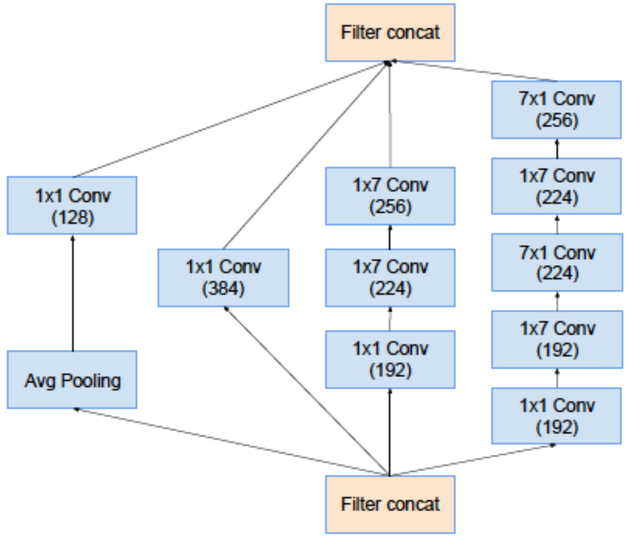
# InceptionV4: Inception A-C Block

CUPOY

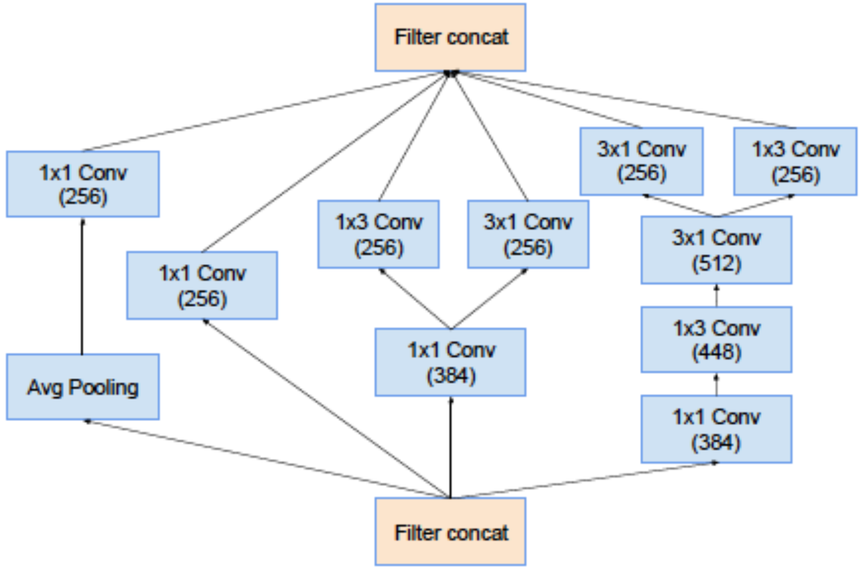
## Inception-A Block




## Inception-B Block




## Inception-C Block

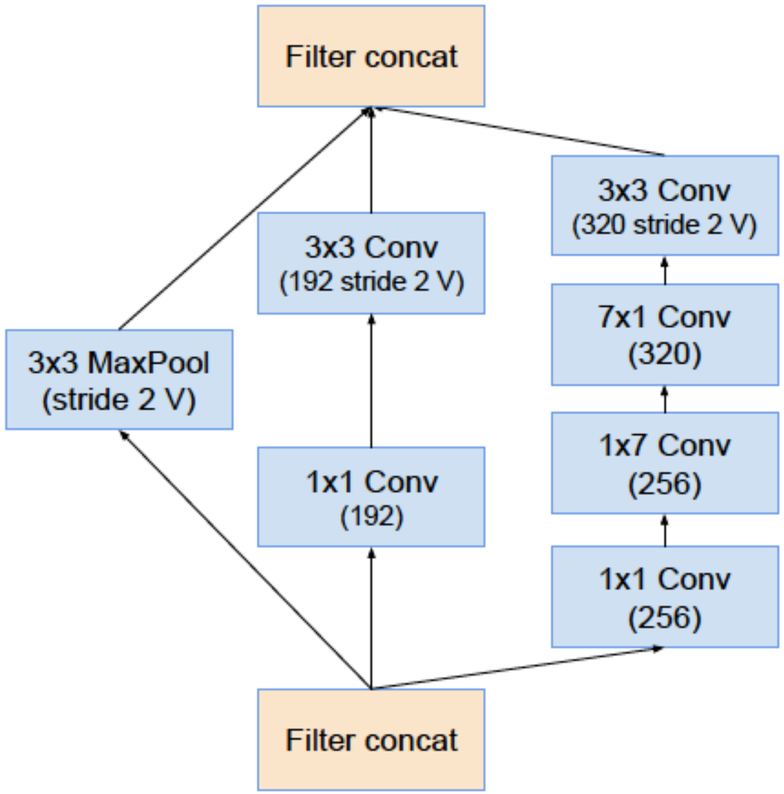


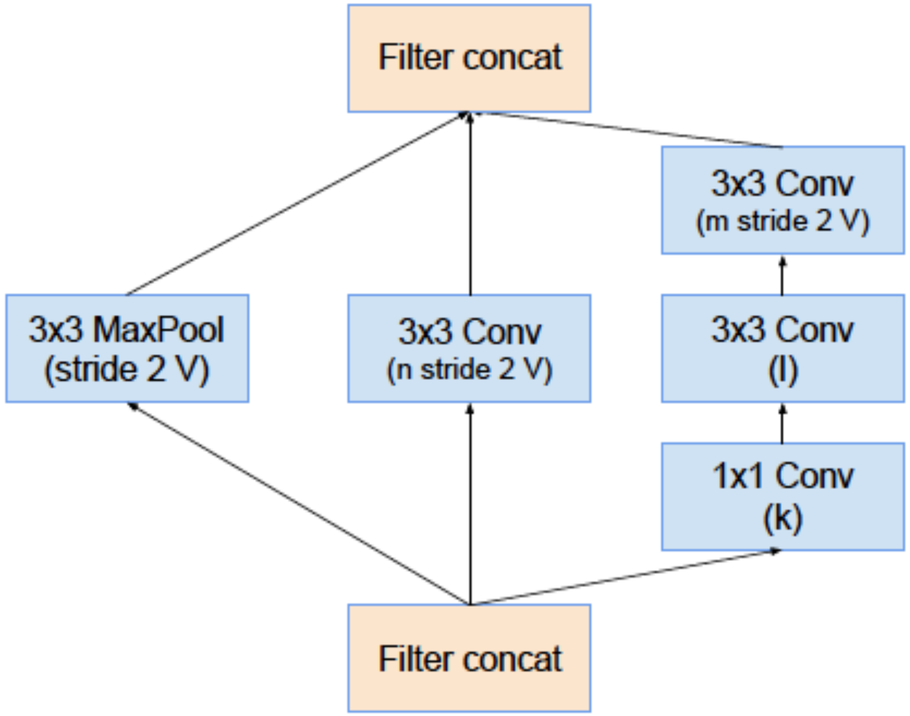

**InceptionV4 的 Inception Block 結構更為多變，堆疊深度也較先前版本來得更深。**

參考來源： Arxiv

# InceptionV4- Reduction Block

Reduction Block 使用 Stride=2 使 Feature Map 尺度下降。

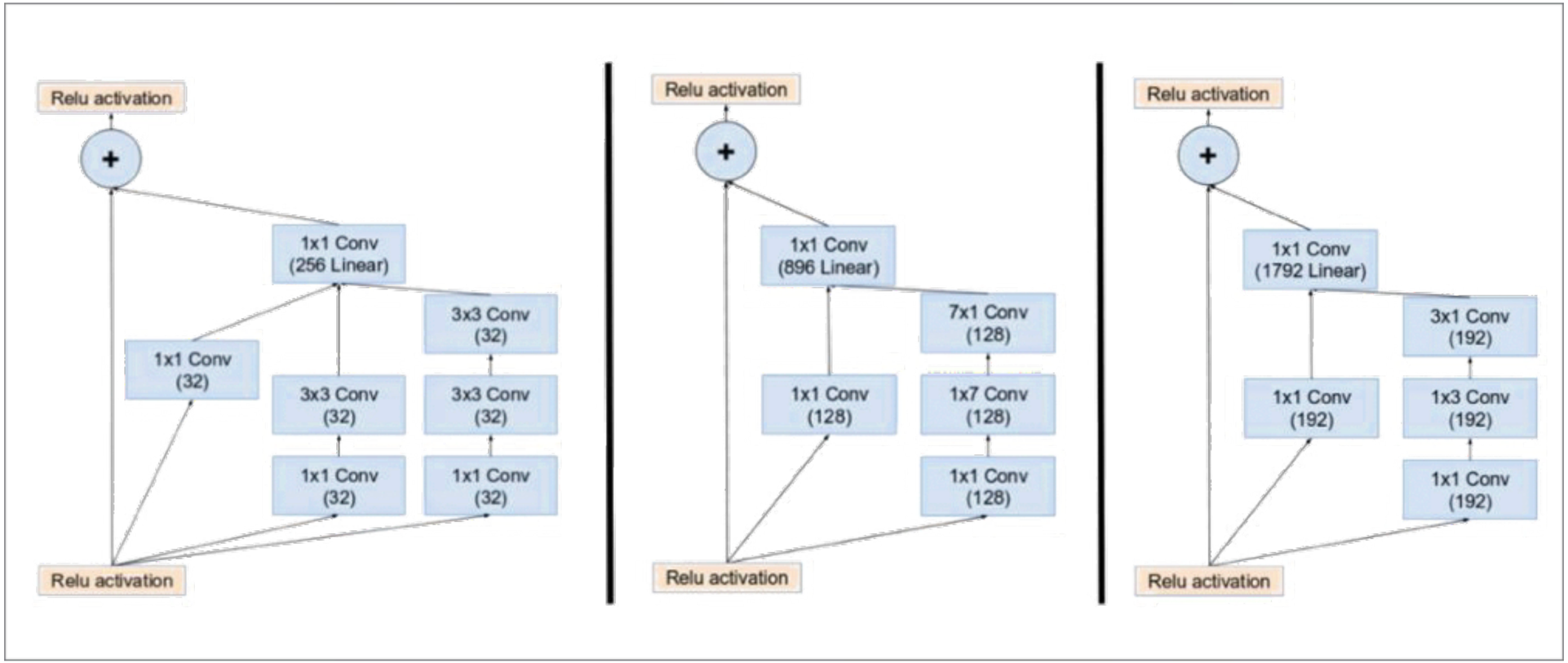



參考來源： Arxiv

# Inception-ResnetV1 : Inception A-C Block

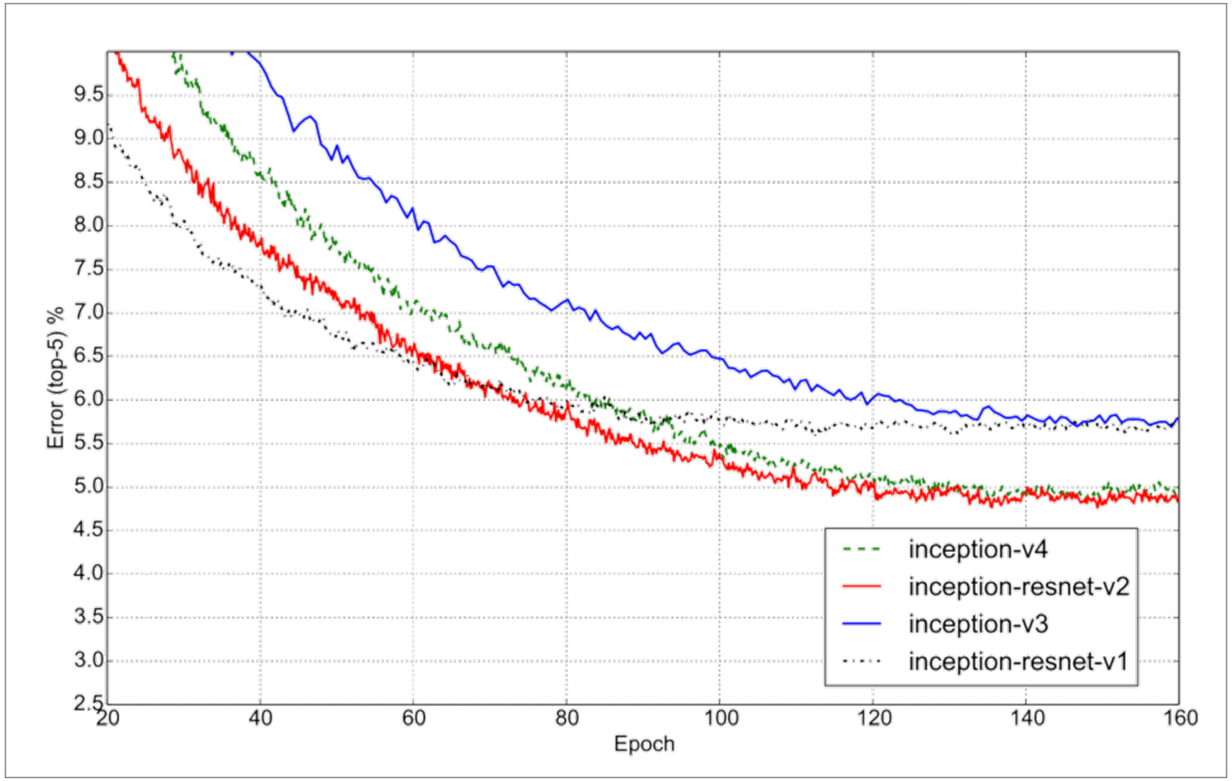
Inception-Resnet版本導入了**殘差網路**的結構，使模型深度能更深。




參考來源： Arxiv

# Inception-ResnetV1 : Inception A-C Block

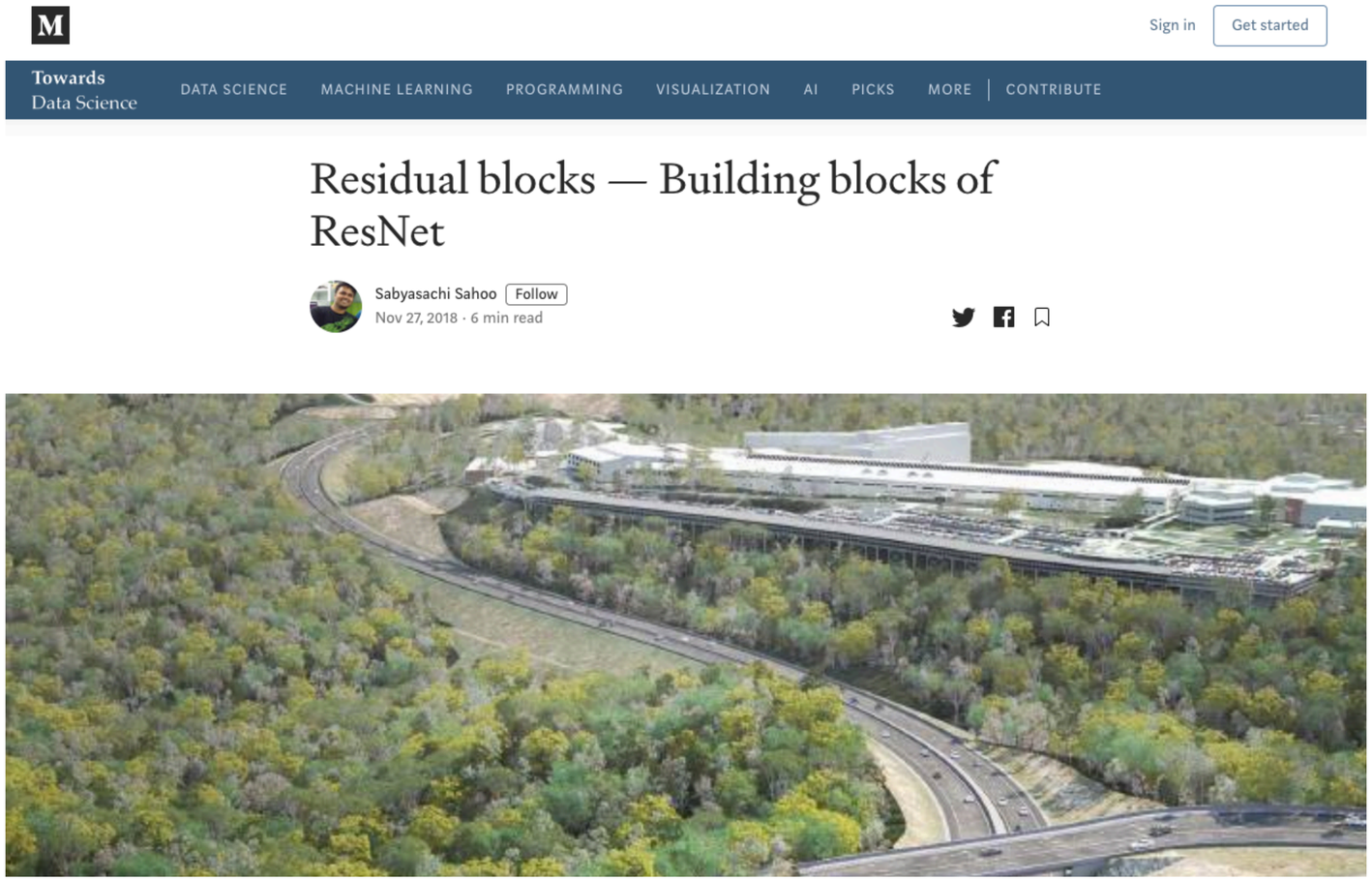
- 可以發現有加入**殘差結構**的版本收斂速度都大幅提升了許多。
- 論文中也有提到，**相同的參數量**下加入殘差結構**並不一定能提升準度**，但確實能**提升模型收斂速度**，並且能搭建**更深的網路**。




參考來源： Arxiv

# 推薦延伸閱讀

CUPOY




**殘差概念**

連結

# 解題時間 Let’s Crack It

CUPOY

請跳出 PDF 至官網 Sample Code & 作業開始解題