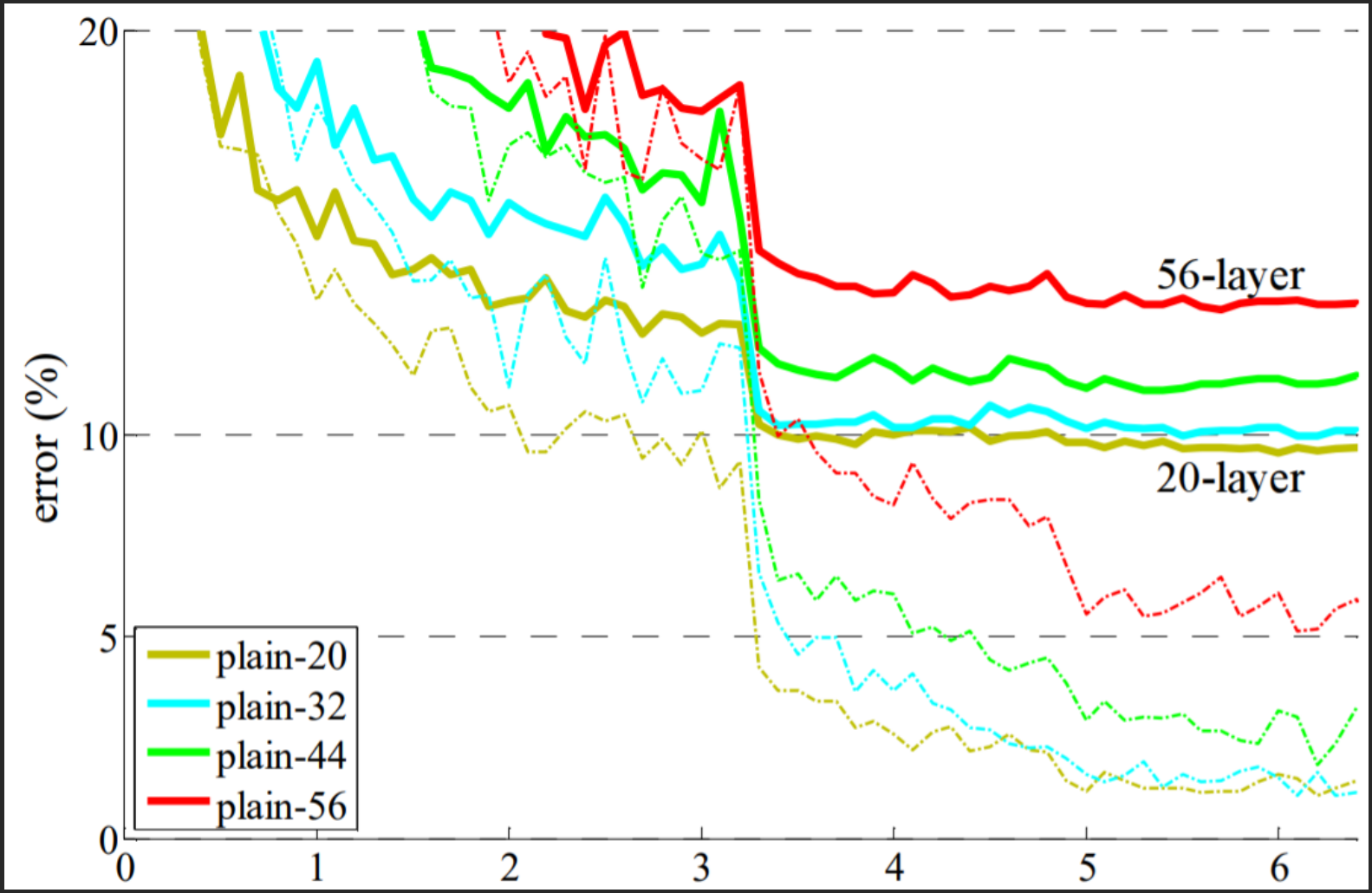
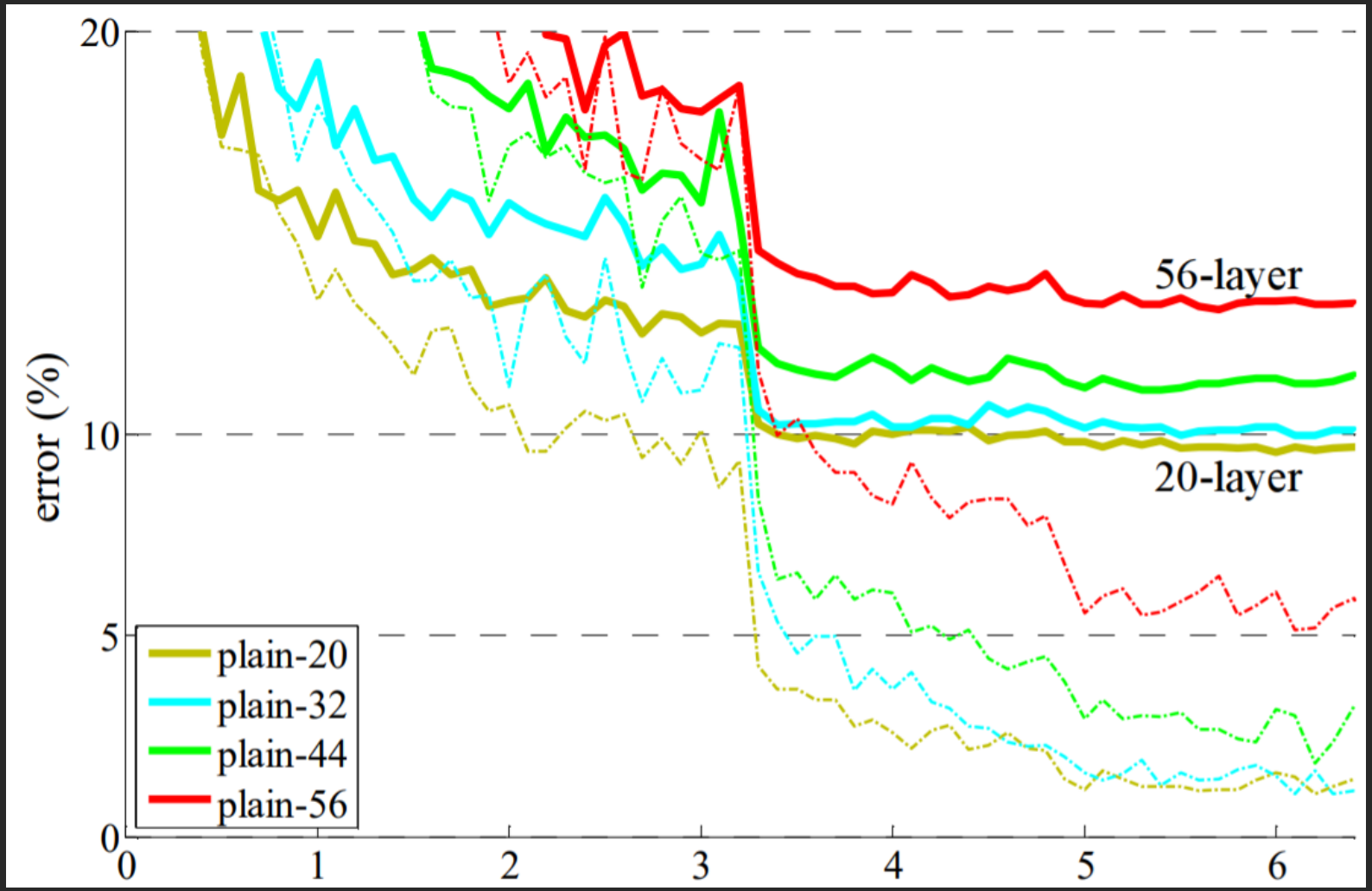
關於 ResNet 的三兩事

問題：模型是不是越深越好

答案是否定的

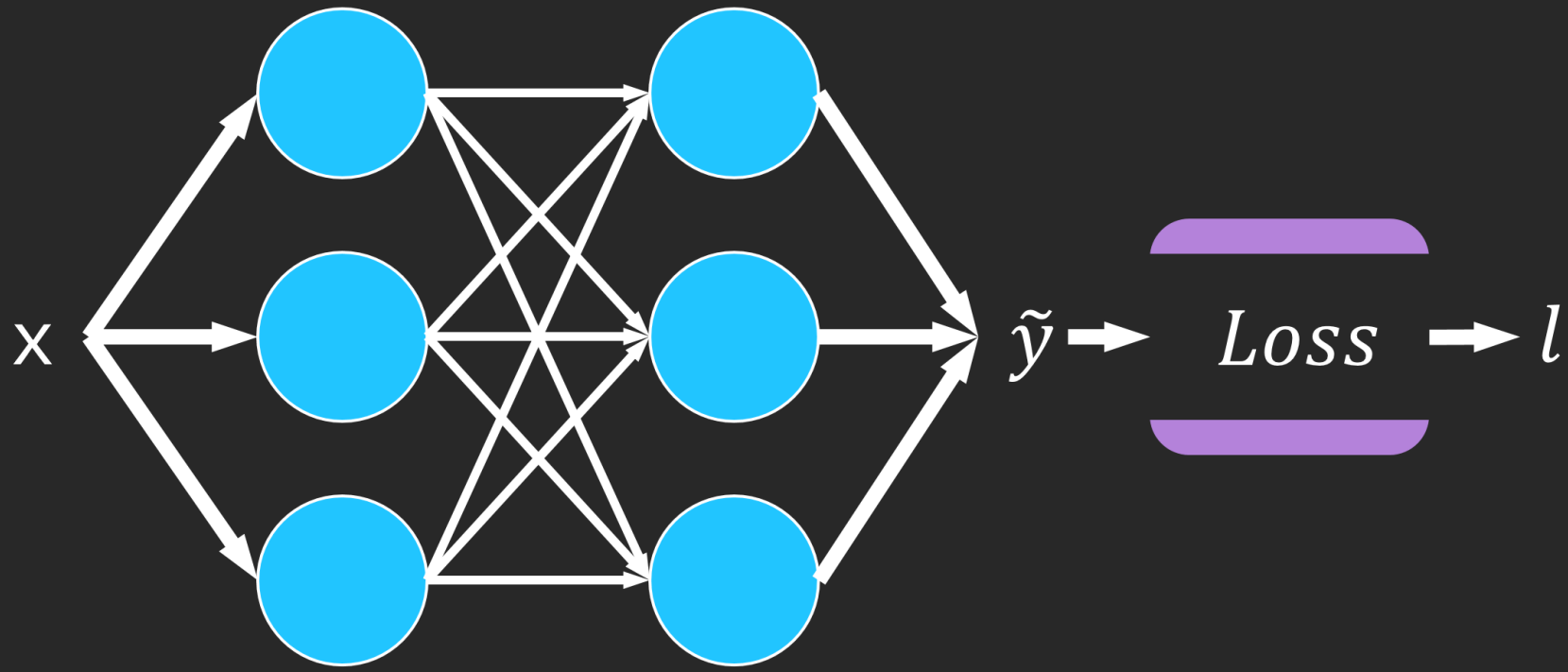

20
10
5
0
error (%)
0
1
2
3
4
5
6
56-layer
20-layer
plain-20
plain-32
plain-44
plain-56

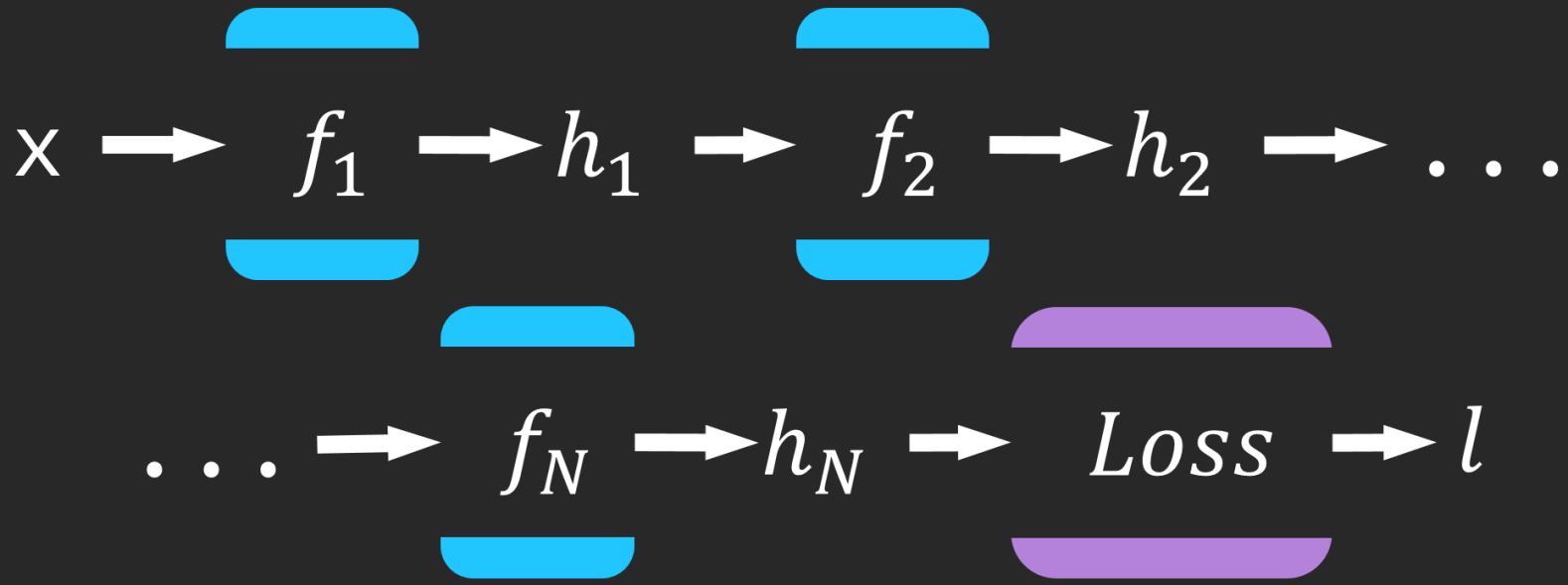
可以看到，增加層數反而使 error 增加了



這個現象是源於

梯度消失
Vanishing Gradient


x
$\tilde{y}$
Loss
l



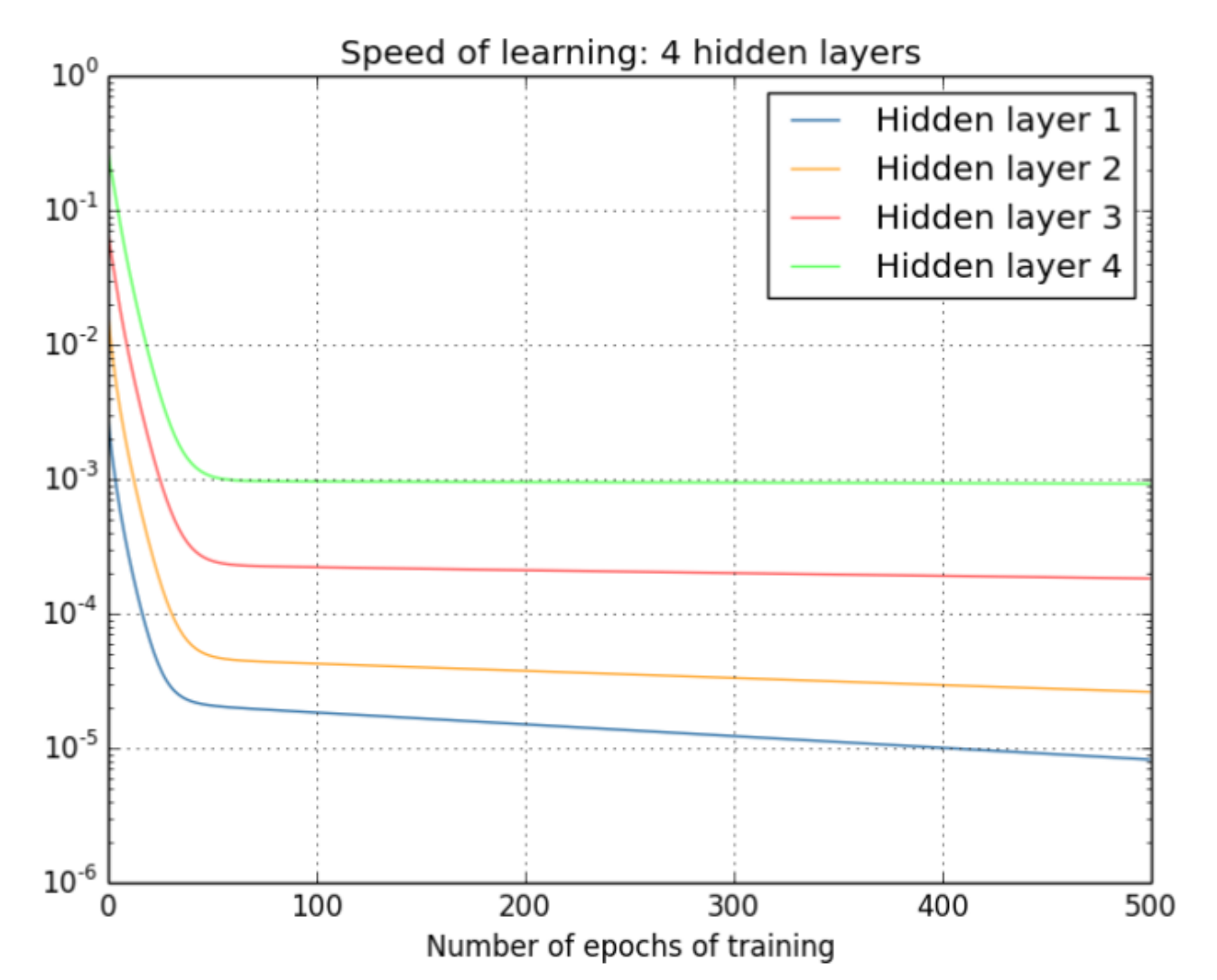
$$h_i = f_i(h_{i-1}) = \alpha(w_i h_{i-1} + b_i)$$

$$\nabla w_i = \frac{\partial l}{\partial w_i} = \frac{\partial l}{\partial h_N}\frac{\partial h_N}{\partial h_{N-1}} \cdots \frac{\partial h_{i+1}}{\partial h_i}\frac{\partial h_i}{\partial w_i}$$

$$\nabla w_i = \frac{\partial l}{\partial w_i} = \frac{\partial l}{\partial h_N} \underbrace{\frac{\partial h_N}{\partial h_{N-1}} \cdots \frac{\partial h_{i+1}}{\partial h_i}} \frac{\partial h_i}{\partial w_i}$$

越多項越容易出現梯度消失or爆炸

越靠前面的 layer 更新越慢

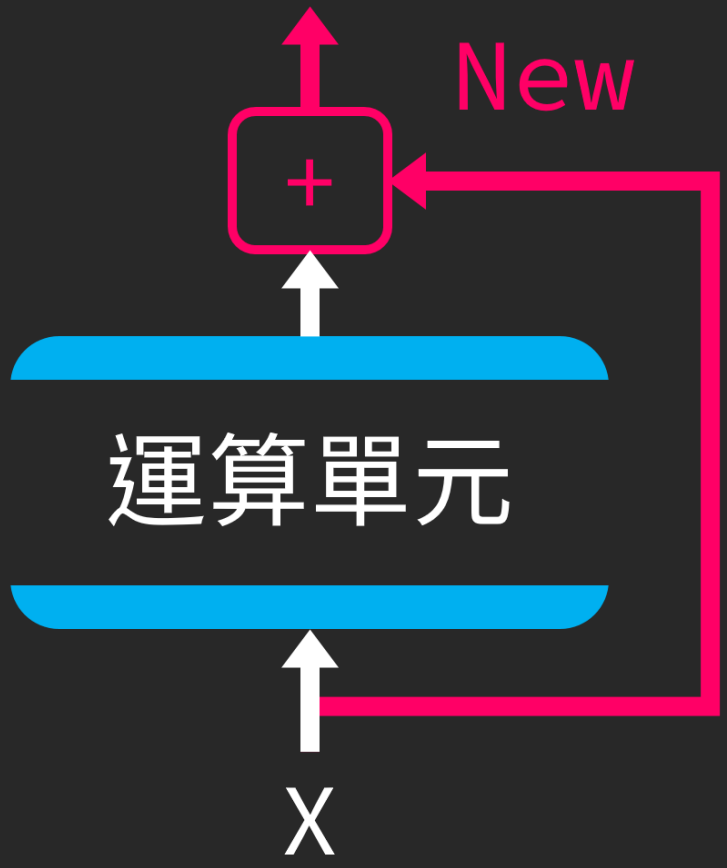


超有效又簡單的解決方案

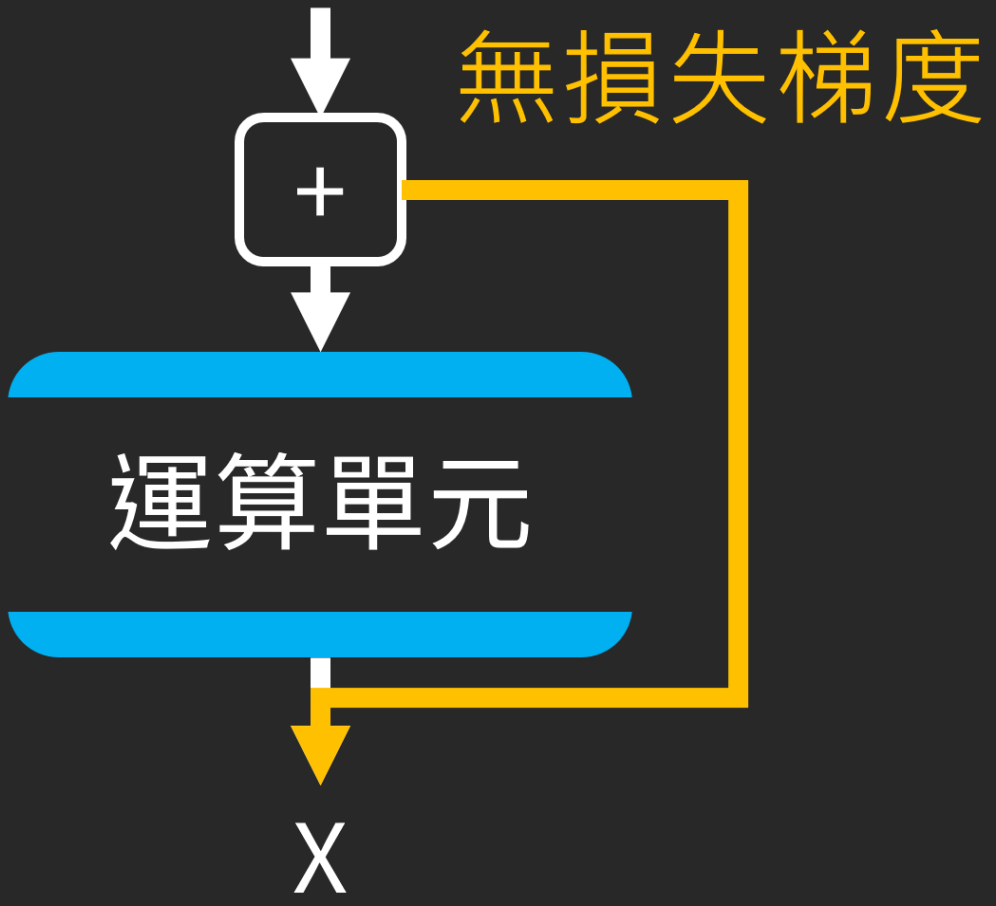
Residual Block

白話文：加一個 shortcut

Shortcut Forward



Shortcut Backward



shortcut 達成了

恆等映射
Identity Mappings

消除了冗於結構的負面影響

恆等映射
Identity Mappings

$$f(x)=x$$

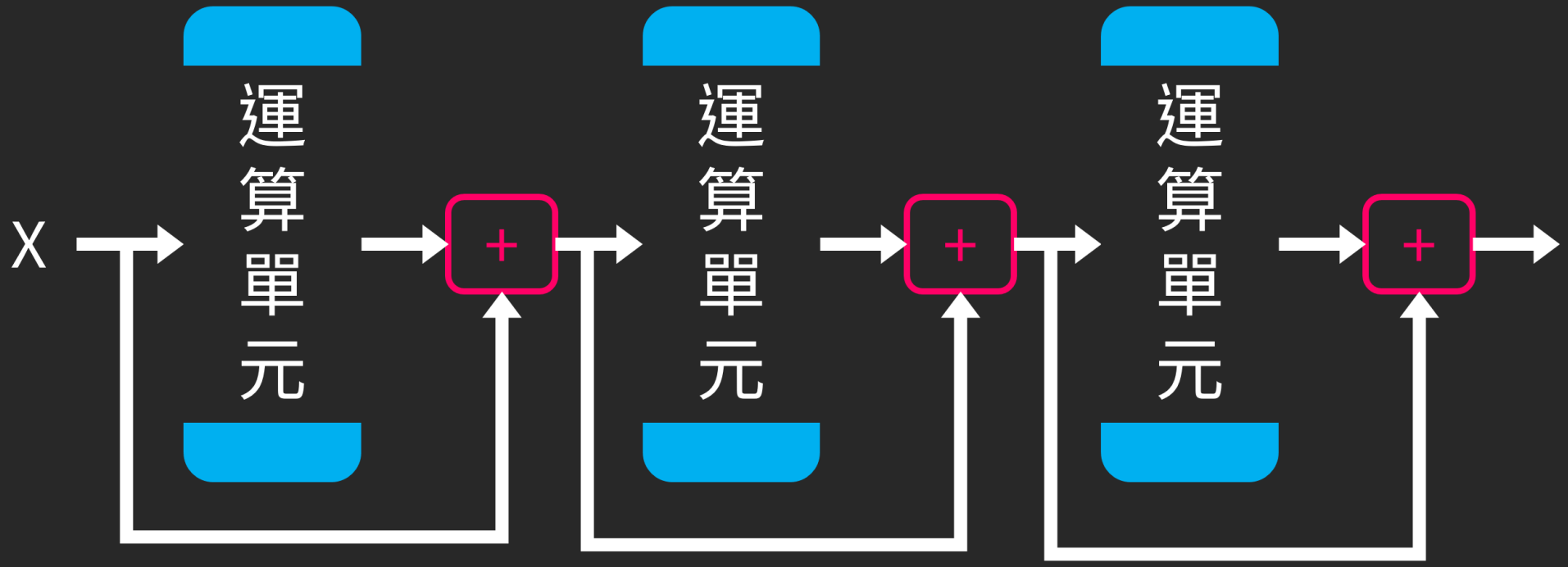
$$f(x)=g(x)+x$$

如果$g(x)$對解決問題是沒有幫助的

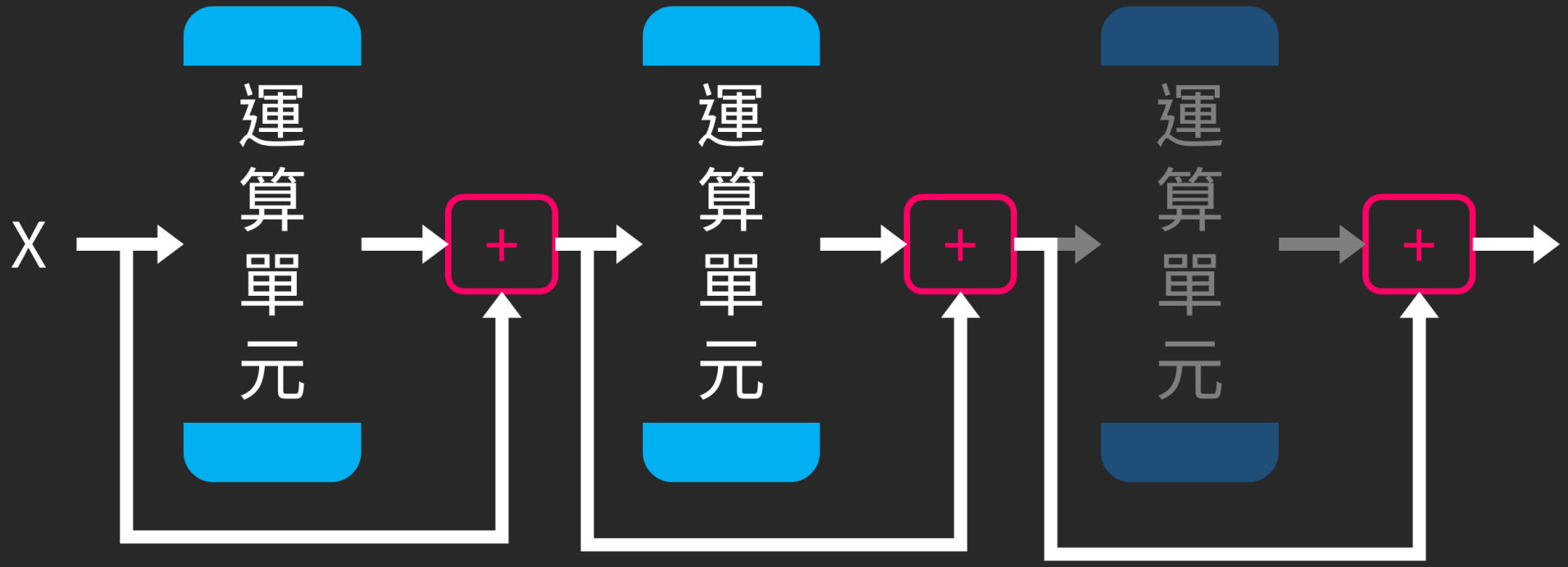
那就讓它輸出為 0

$$f(x)=g(x)+x$$

人為設計的多層結構，不一定每一層都有用處



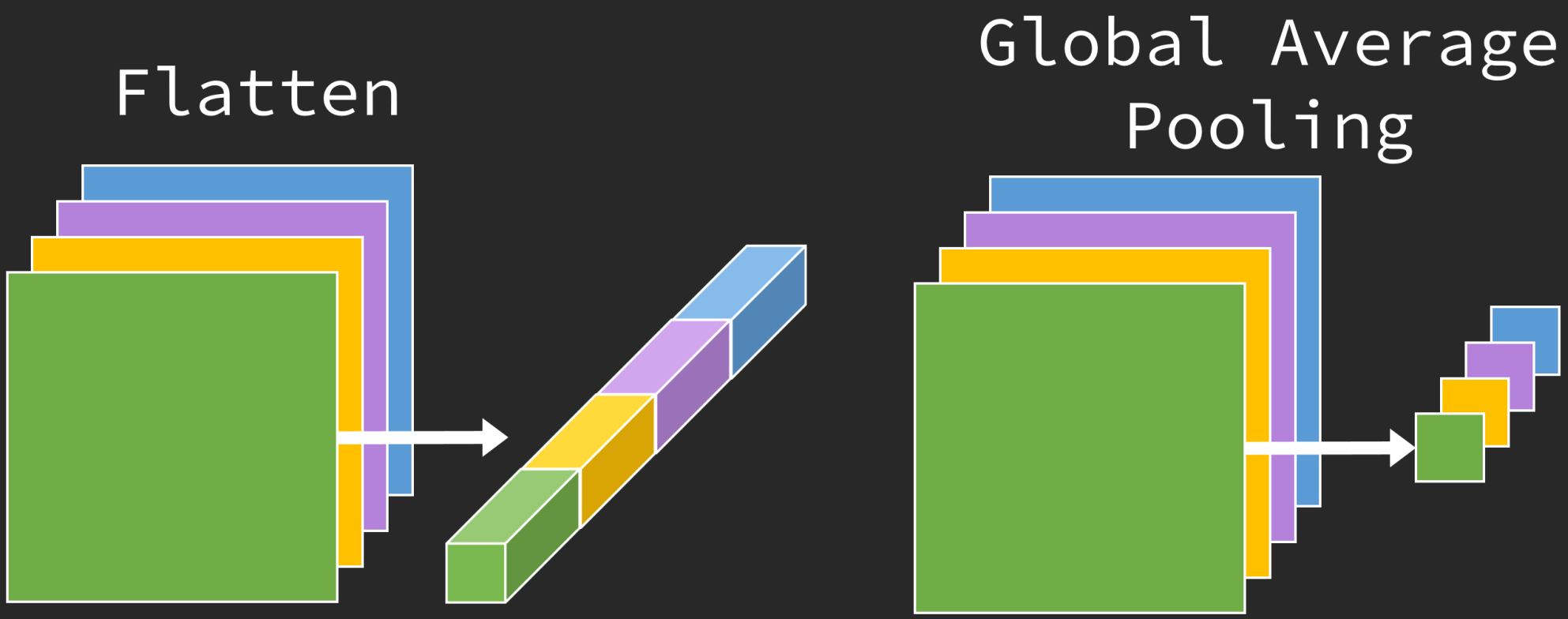
模型只要將冗於的部分歸 0 就不會受到拖累



ResNet 官方結構

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

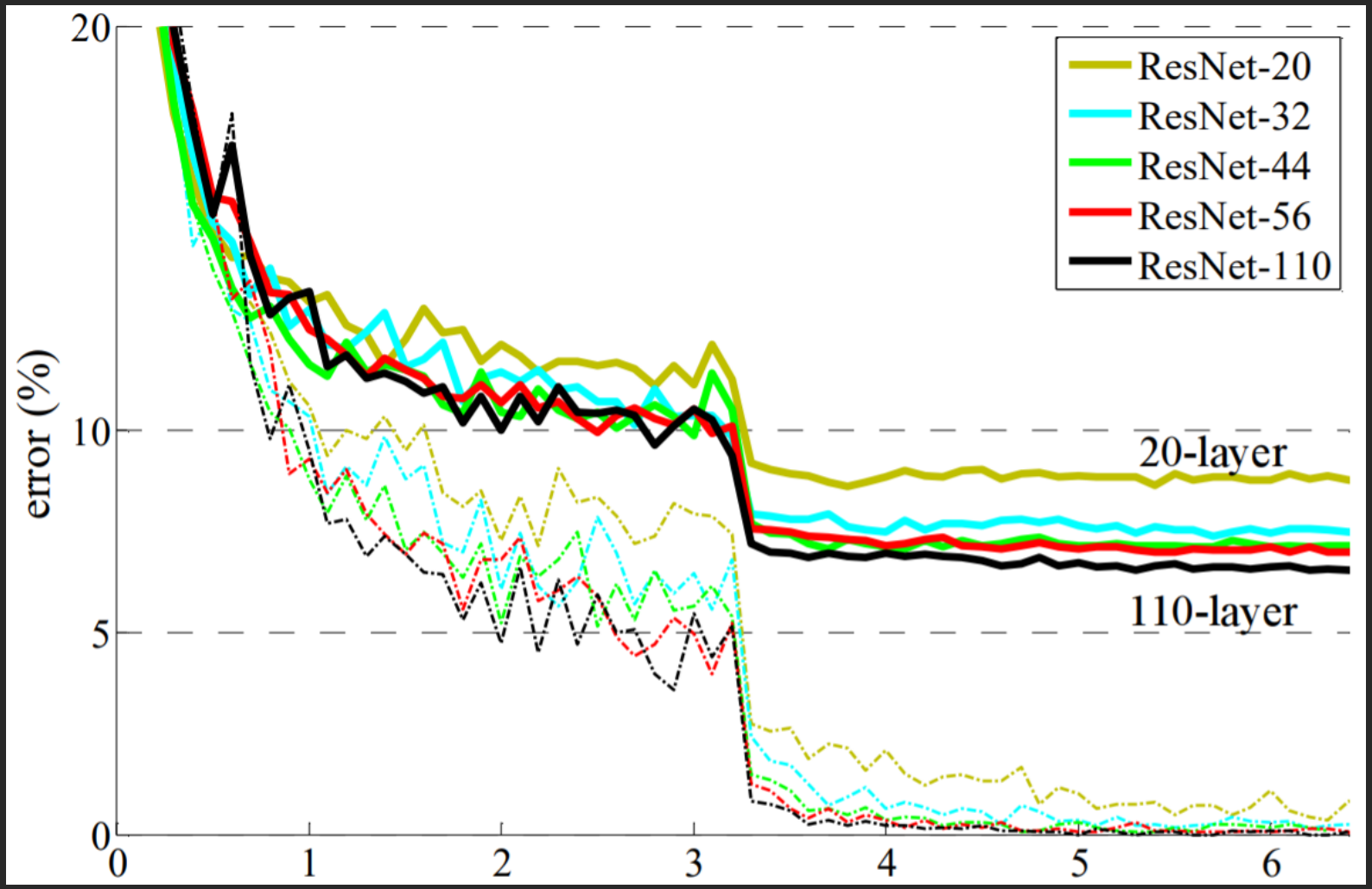
Global Average Pooling



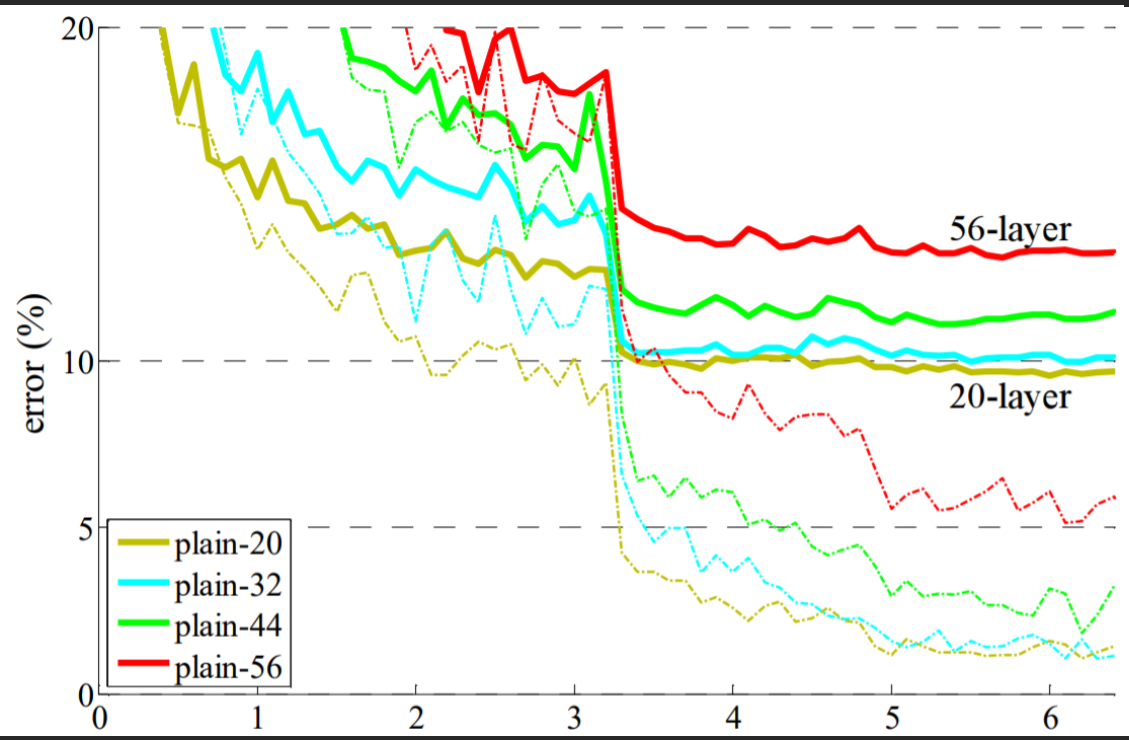
優點

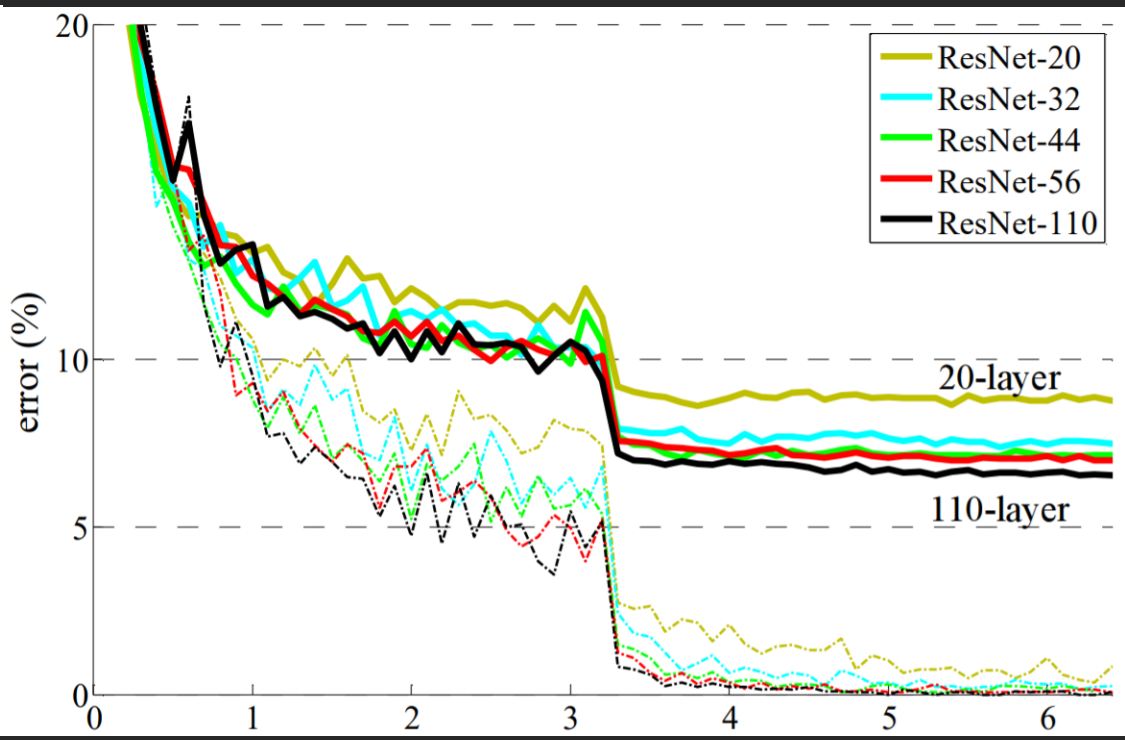
- 減少參數
- 避免 overfitting
- 可變動輸入大小

ResNet 效果


ResNet-20
ResNet-32
ResNet-44
ResNet-56
ResNet-110
20-layer
110-layer
error (%)
20
10
5
0
0
1
2
3
4
5
6

PlainCNN vs ResNet





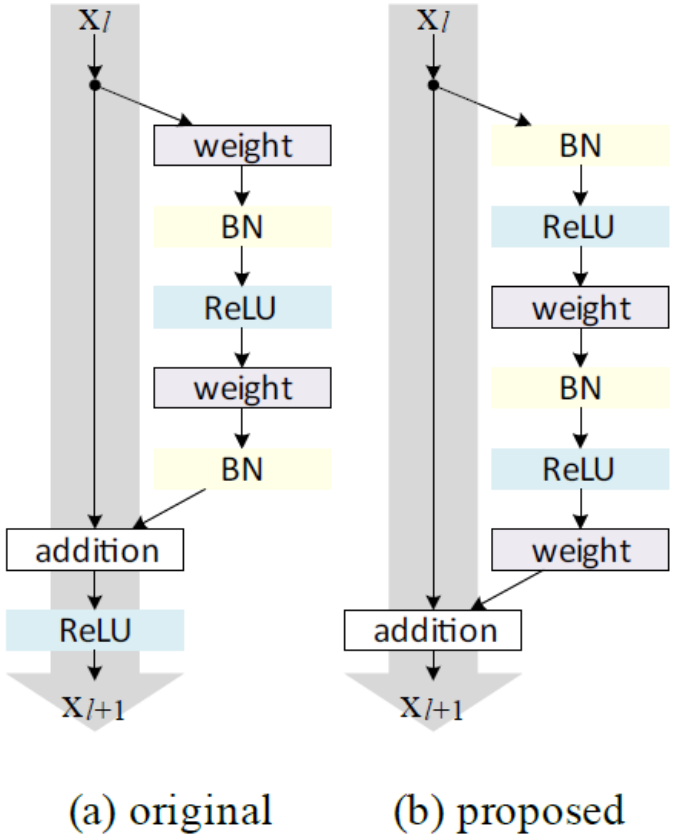
Shortcut 越乾淨越好

在 Identity Mappings in Deep Residual Networks 進行了許多關於 Shortcut 的實驗

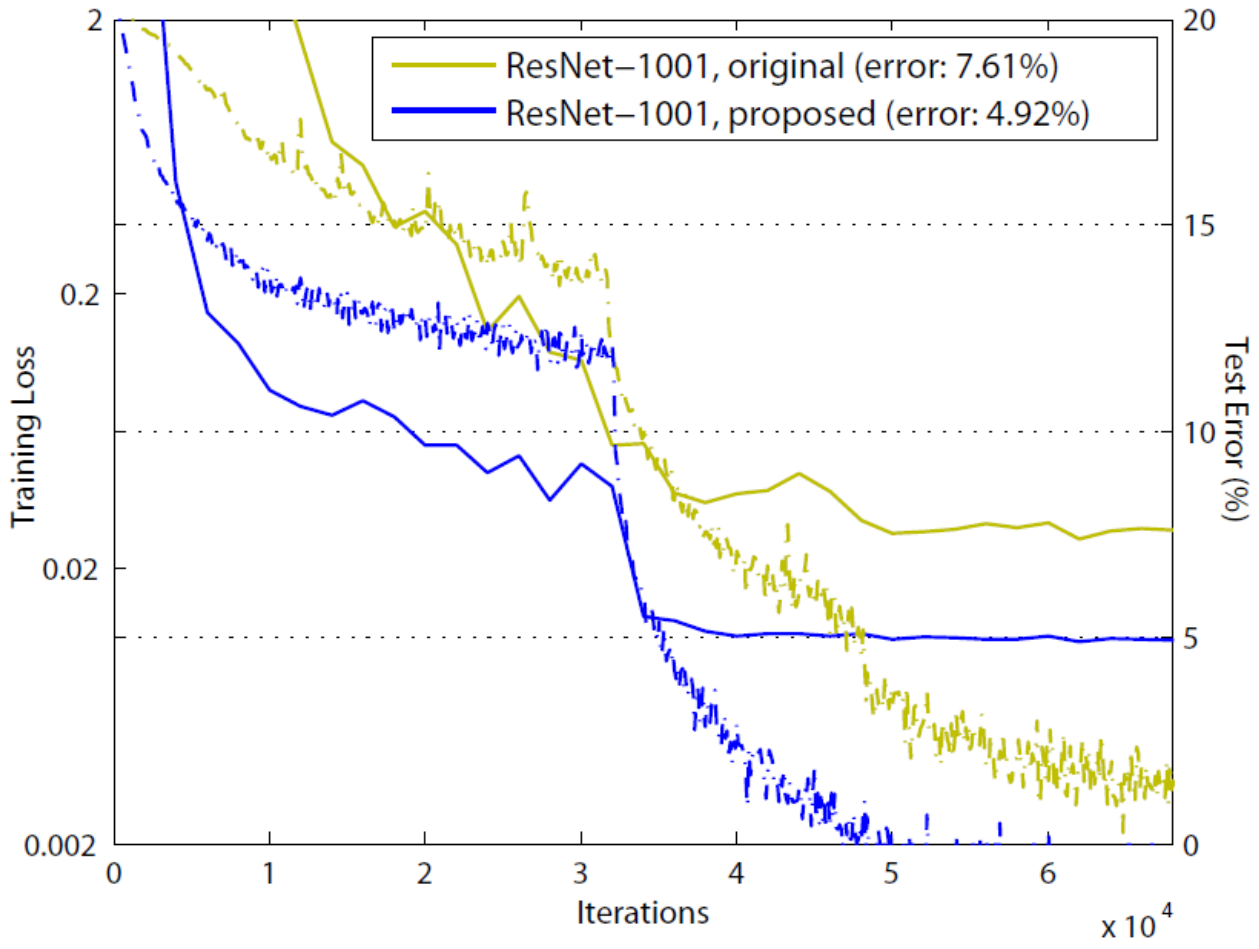
其中最重要的結論就是：Shortcut 不要有多餘運算

讓梯度能利用這條捷徑不受任何影響的往回傳

Shortcut 越乾淨越好



(a) original (b) proposed



更多詳細的實驗請看

Deep Residual Learning for Image Recognition

與

Identity Mappings in Deep Residual Networks

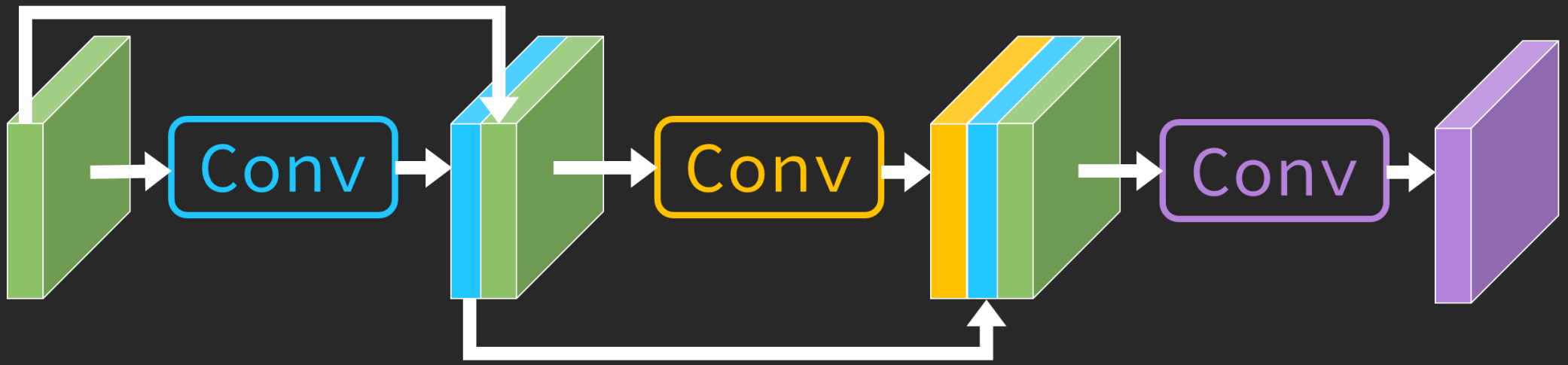
参考

Identity Mappings in Deep Residual Networks

論文介紹

加碼：更多種 Block

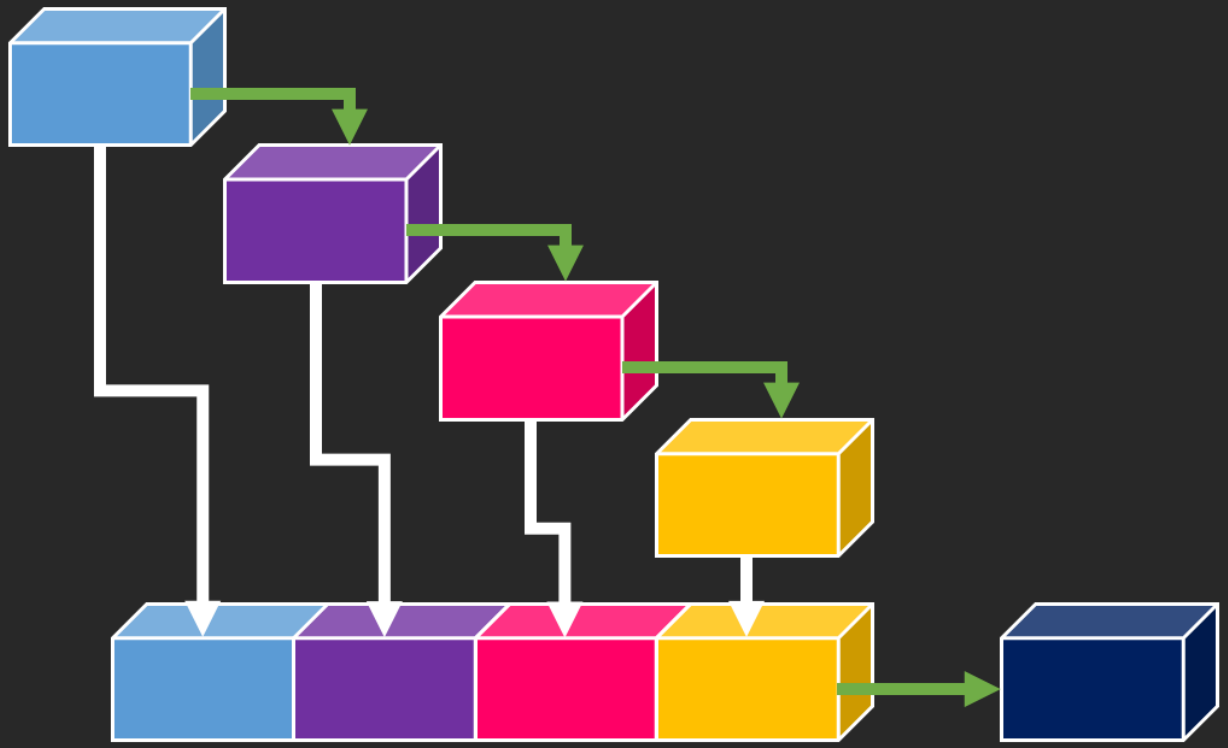
DenseNet



DenseNet

- 除了做到恆等映射外，還充分整合了各種感受野的特徵。
- 只要更少的權重與計算量就能做的比 ResNet 更好。

VoVNet



VoVNet

- 繼承 DenseNet 的優點。
- 比 DenseNet 更快速輕量。

小挑戰

- 請規劃一個實驗，來證明模型是不是越深越好。
- 請規劃一個實驗，向大家說明 shortcut 是否有效。
- 請比較 residual block 與另外一種 conv block 的效果。