中華大學生物資訊學系系統開發專題報告

# 探討批次正規化對深度學習影像分類之效能影響

# Investigation of batch normalization effect on the performance of deep leaning image classification

專題組員:陳曉柔、陳珮姍

指導老師:黃俊燕老師

## 1. 摘要

本專題利用 Cifar-10數據集，來探討批次正規化對深度學習影像分類之效能影響，首先我們探討不同的學習速率、批次大小、活化函數以及批次正規化與活化函數在模型中的擺放順序，對模型訓練結果的影響。使用自行設計的四種模型進行訓練，我們所獲得的結果顯示，在使用批次正規化時，可以使用較大的學習速率進行訓練，但在批次大小過小時，批次正規化會使得模型不易學習到特徵分佈，導致訓練的模型泛化能力降低，對於所有四種自行設計的模型，批次正規化層擺放在活化函數層之前皆優於擺放在其後。

關鍵詞: 學習速率、批次大小、活化函數、批次正規化。

## 2. 簡介

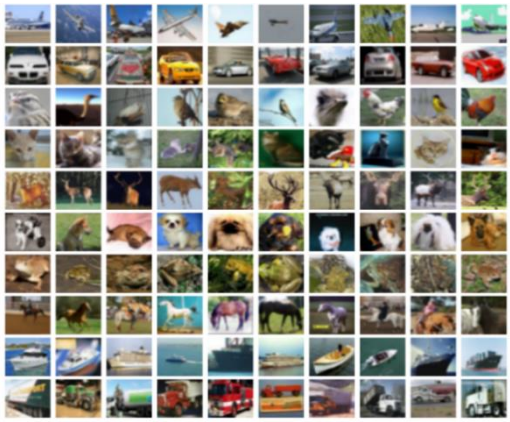
深度學習開始於20世紀40年代，1943年，McCulloch和Pitts提出了MP模型[1]，這是使用閥值邏輯運算法， LeNet5 誕生於 1994 年，是最早的卷積神經網絡之一[2]，大約在2006年時，Hinton 等人他們使用逐層初始化的方式來訓練包含很多個隱藏層的網路，在隨後幾年在語音辨識和影像辨識領域有了巨大成功，因此神經網路正式稱為深度學習，在2012年的ImageNet 比賽中，AlexNet 網路的功能大幅超過傳統的機器學習方法[3]，此時真正的深度學習時代才開啟，2014年牛津大學研發出的VGGNet網路[4]，證明了加深網路能夠影響模型最終的一般化能力提升，在2014年出現了 Network in Network，可以在訓練模型時，使參數減少[5]，在同年 Google 的 Chrisrian Szegedy 設計出GoogLeNet [6]，第一個 Inception 架構，擷取不同尺度大小的特徵，進行合併，在2015年何凱明提出深度殘差學習(ResNet) [7]，採用了跨層連接方式，可以緩解神經網路中梯度消失的問題，在同一年，康奈爾大學、清華大學和 Facebook 聯合提出的 DenseNet [8]，能強化特徵得傳遞和特徵重複使用，並減少參數。

在深度學習中，往往會將網路層數加深進行訓練，會造成梯度消失。批次正規化(Batch Normaliztion，簡稱：BN) [9]是將每一層卷積層(Convolution Neural Networks，簡稱：CNN)輸出的特徵先做平均值為0、標準差為1的操作，可以避免梯度爆炸與梯度消失的問題，可以使用較大的學習速率從而加速了模型的學習收斂，由於批次正規化也具有權重正則化的功能，同時也可解決過度擬合增進模型泛化能力。

3. 專題進行方式

3.1 資料來源

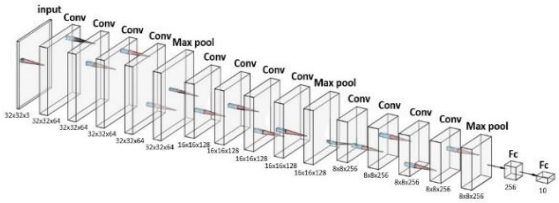
資料取自於 Cifar-10數據集[10]，此數據集中有60000張32x32大小的彩色圖像，其中50000張為訓練資料，再從訓練資料中取出5000張做為測試資料，分別有10種類別的照片，有 airplane、automobile、bird、cat、deer、dog、frog、horse、ship、truck，每一個類別各有5000張。



圖一 Cifar-10 數據集

3.2 模型設計

影像分類主要先由卷積層(Convolution Neural Networks，簡稱：CNN)用來擷取特徵，再接全連接層做分類。而我們設計的模型可分為三個區塊(如圖二)，第一個區塊有4層卷積層，每一層 CNN 層使用的活化函數為LeakyReLU，其$\alpha$值為0.01，每一層卷積核均為64個，且尺寸為3x3、移動步長(stride)為1，末尾為一個卷積核尺寸為2x2、移動步伐為2的最大池化層，每一個 CNN 層之卷積核個數皆相同，第一個區塊的卷積核數為64；第二個區塊的卷積核數為128；第三個區塊的卷積核數為256，呈現兩倍數的指數增長，卷積層獲得的特徵地圖展平後，輸入至256個神經元的全連接層，再使用丟棄層(Dropout)，每一批次隨機選擇50%的神經元使其不放電，最後輸出層為10個神經元的全連接層做分類，此模型可簡稱為64K-4C3B-256D-Dropout0.5-10D。




圖二 模型圖

3.3 資料擴增：

為了解決資料不足而導致學習能力過度擬合的問題，所以將訓練資料的數量變為原始資料的3倍，並把照片做些變換，製造不同的影像，其改變包含有隨機旋轉-70到70度(如圖三-2)、隨機水平平移-10%到10% (如圖三-3)、垂直平移-10%到10% (如圖三-4)、隨機水平翻轉(如圖三-5)。

(1)原始影像 (2)隨機旋轉

(3)水平位移 (4)垂直位移 (5)水平翻轉

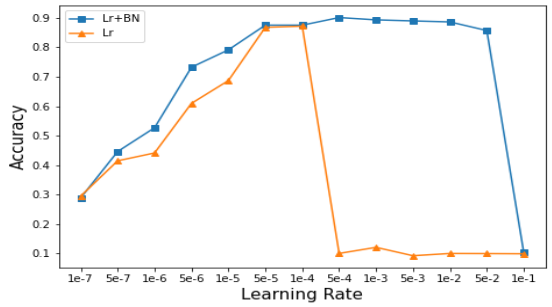
圖三 資料擴增

4. 主要成果

4.1 學習速率與模型訓練關聯性

在我們設計的這個模型，使用 Adam 的最佳化演算法，以訓練週期為60進行模型的訓練，測試不同的學習速率所能獲得的測試資料精確度。當學習速率太大時，梯度下降時，可能會越過低點，導致無法收斂甚至發散；學習速率太小時，收斂速

度太慢，會導致學習速度過慢，並且容易侷限在局部極小值，導致一般化能力不佳。加了批次正規化後，學習速率最大可以使用到0.05，學習速率可用範圍從0.000005到0.05，比沒有批次正規化的學習速率的使用範圍變廣也變大，有效地使損失函數平面變的平滑，比較不易被侷限在局部極小值無法逃脫，因此訓練好之模型具有較佳的一般化能力(如圖四)。

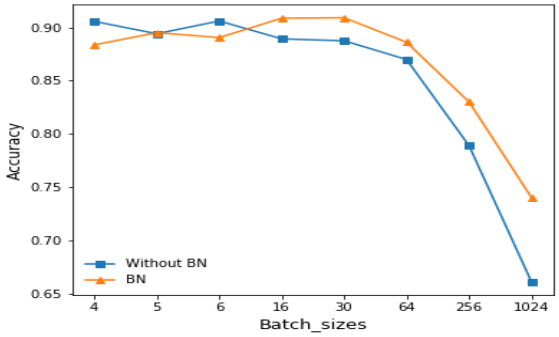



圖四 學習速率

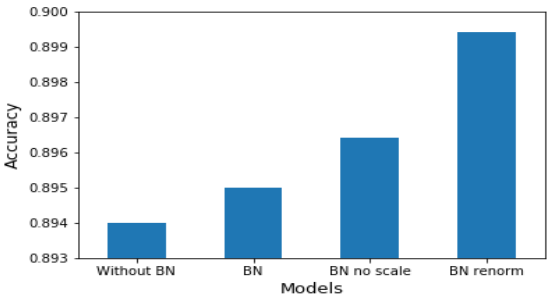
## 4.2 批次大小對模型訓練結果的影響

由於批次訓練法訓練一個模型需要使用全部的訓練資料，資料量與計算量過大可能超過 GPU 硬體資源而導致無法訓練，小批次學習法是將訓練資料分成若干組的小批次資料，每一組小批次資料內的資料數量稱之為批次大小(Batch size)，再依每一個小批次組中的資料來訓練模型。每一個學習週期時，計算出每一個小批次的損失函數平均值，依此計算出損失函數梯度並調整網路權重參數，直到搜尋到損失函數最小值之權重參數。在做批次大小測試時，我們使用自行設計的模型、階梯函數學習速率調節(Learning Rate Scheduler)、Adam 最佳化演算法、以訓練週期為60進行模型的訓練，階梯函數學習速率調節(Learning Rate Scheduler)的初始學習速率為0.0001，到了第21個訓練週期時，把學習速率改為0.00005，在第31個訓練週期時，把學習速率改為0.00001，在第41個訓練週期時，把學習速率改為0.000005，在第51個訓練週期時，把學習速率改為0.000001。

當批次大小較大時，學習收斂速度較快可以減少訓練的時間，由於每一個小批次估算的損失函數較為平滑，使得最佳化演算法無法調整至夠低損失函數值的權重參數，從而導致模型的一般化能力降低；反之，批次大小較小時，訓練時間會較長，但可以搜尋到損失函數極小值附近的權重參數，提升模型的一般化能力。




圖五 批次正規化對批次大小之影響。Without BN：沒有批次正規化。BN：批次正規化。




圖六 調動批次大小的正規化方式。Without BN：沒有批次正規化。BN：批次正規化。BN no Scale：批次正規化不除標準差。BN renorm：批次重新正規化。

訓練卷積網路模型加速收斂提升精確度常用的技巧為批次正規化(Batch Normalization)，所謂批次正規化是在每

一個小批次訓練當中，將每一層卷積網路層的輸入資料調整為平均值0標準差為1的分布，假設$x_i$代表卷積層的輸入資料，n為批次大小，輸入資料的小批次平均值為，

$$\mu_B = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (1)$$

輸入資料的標準差為，

$$\sigma_B^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_B)^2 \quad (2)$$

對每一層卷積網路層的輸入資料做正規化，ε是為了防止分母為0的一個非常小的常數。

$$x_i' = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \quad (3)$$

加了批次正規化後，批次大小低於4時，因為批次大小太小，導致在做批次正規化時，卷積層輸入資料於不同小批次的平均值和標準差差異較大，因此導致不同小批次計算之梯度值大幅度的漲落，在學習訓練時較沒效率，往往錯過損失函數極小值，所以一般化能力較沒有加入批次正規化訓練的結果差(如圖五)。為了解決此問題，我們讓資料只平移至平均值為0，而不去調整資料標準差的尺度，其結果變為0.8965，較未修正前0.8950佳。除此之外，我們再嘗試第二種修正方式批次重新正規化(Batch Renormalization)[11]，批次重新正規化採用移動平均與標準差取代小批次平均與標準差，增加了兩個仿射變換參數$\gamma$和 $d$，並且將結果修正至0.8995(如圖六)。

批次重新正規化(Batch Renormalization)公式

$$\frac{x_i - \mu}{\sigma} = \frac{x_i - \mu_B}{\sigma_B} \cdot \gamma + d \quad (4)$$

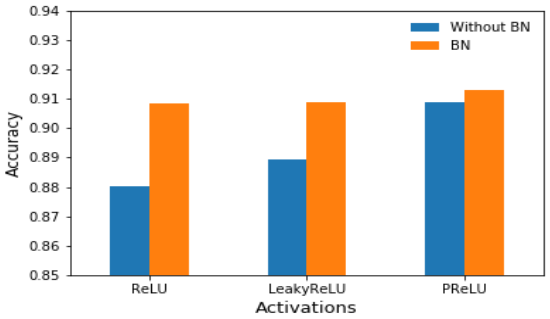
參數γ和$d$，

$$\gamma = \frac{\sigma_B}{\sigma},\ d = \frac{\mu_B - \mu}{\sigma} \quad (5)$$

其中μ為移動平均值σ為移動標準差。

### 4.3 不同活化函數對模型訓練結果之影響

非線性活化函數是卷積網路能夠進行影像分類辨識的關鍵，不同的活化函數會使模型一般化能力有所不同。在做活化函數測試時，我們使用相同的階梯函數學習速率調節(Learning Rate Scheduler)、最佳化演算法、以訓練週期為60進行模型的訓練。使用 ReLU 時，可以解決使用 Sigmoid 活化函數的梯度消失問題，只需要判斷輸入值是否大於0即可，收斂速度較快，但會導致部分神經元永遠不會放電參與計算，使得其相對應的參數無法被更新學習，這個現象稱之為神經元死亡問題(dead node problem)，LeakyReLU 活化函數可以解決 ReLU 的缺陷，而被提出來，要判斷小於0時，在乘上一個$\alpha$值，解決了神經元死亡問題，PReLU 是 LeakyReLU 的改良，$\alpha$值是可以訓練的參數。加了批次正規化後，每一層的輸出值正規化到平均值0標準差為1的分布，也確保梯度有效，且不會有梯度爆炸的問題，所以活化函數的效應不明顯。在有使用批次正規化時，活化函數不管使用 ReLU、LeakyReLU 還是 PRelu，都會比沒有利用批次正規化時的模型一般化能力變佳，使用活化函數 ReLU 的測試資料精確度上升到0.9082，活化函數 LeakyReLU 的測試資料精確度上升到0.9086，活化函數 PReLU 的測試資料精確度上升到0.9128，所以可以得知在活化函數 ReLU 時，最能體現出批次正規化的作用。(如圖七)




圖七 批次正規化對不同活化函數之影響

ReLU 公式

$$f(x) = max\{0, x\} \tag{6}$$

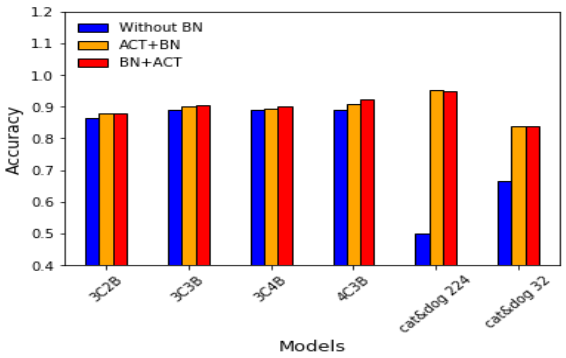
LeakyReLU 公式

$$f(x) = \begin{cases} \alpha x, x \leq 0 \\ x, x > 0 \end{cases} \tag{7}$$

我們使用的α值為0.01，對於較大的α值，例如0.3，將導致模型訓練的結果泛化能力較差。

### 4.4 批次正規化與活化函數順序之效應

提出批次正規化的作者 Sergey Ioffe 與 Christian Szegedy，於該論文中是將批次正規化加在活化函數前[10]，然而許多實證經驗顯示，使用批次正規化時，如果將其加在活化函數之後，一方面也較為合乎直覺且能夠獲得較優異的模型訓練結果，我們在此將探討批次正規化與活化函數的擺放順序對影像分類結果造成的影響。我們探討批次正規化與活化函數的擺放順序對四種不同的模型3C2B、3C3B、3C4B與4C3B 造成的影響(如圖八)，於訓練這四個模型時均使用相同的學習速率調節階梯函數(Learning Rate Scheduler)、Adam 最佳化演算法、與 LeakyReLU 活化函數，使用批次大小16對模型進行60週期的訓練。在圖八中我們可以發現，對於 Cifar-10數據集，對於所有這四個模型，批次正規化擺放於活化函數前的結果皆優於擺在活化函數後的結果，此結果與一般大多數數據集的實證經驗相反，使用 Kaggle Cat and Dog 資料集[12]時，當照片大小為224x224時，批次正規化放在活化函數後面時，測試資料精確度為0.9530，批次正規化放在活化函數前，測試資料精確度為0.9506，批次正規化放在活化函數前精確度較高，照片大小為32x32，批次正規化放在活化函數後面時，測試資料精確度為0.8384，批次正規化放在活化函數前，測試資料精確度為0.8374。




圖八 批次正規化在活化函數前後的影響
ACT+BN：活化函數+批次正規化。BN+ACT：批次正規化+活化函數。cat&dog 224：貓和狗資料集照片大小為 224× 224，cat&dog 32：貓和狗資料集照片大小為32 × 32。

### 4.5 自行設計模型之訓練結果

經由以上的探討與研究，我們使用256K-4C3B-1024D-Dropout0.5-10D 模型，將訓練資料擴增至原本的3倍，並在每一層 CNN 卷積層後加上批次正規化(Batch Normalization，簡稱：BN)，在全連接層分類時也加上 BN，使用學習速率調節(Learning Rate Scheduler)、Adam 最佳化演算法、活化函數為 LeakyReLU，使用批次大小為16進行60週期的訓練，訓練結果之各類別平均精確度為0.9454。

## 5. 結論

在深度學習中，運用批次正規化，可以使用較大的學習速率訓練，加速學習的速度並且在訓練過程中較不易被侷限在局部極小值，在批次大小足夠大時有助於提升一般化能力，但批次大小過小時，將導致一般化能力下降，解決方法有，不調整資料標準差的尺度以及使用批次重新正規化，此外我們發現活化函數與批次正規化於模型中擺放的次序會影響模型的泛化能力，其效應因資料集不同而有所不同。

使用自行設計的模型，資料為原始的3倍，使用學習速率調節、最佳化演算法、活化函數為 LeakyReLU，使用批次大小為16進行60週期的訓練，訓練結果之各類別平均精確度為0.9454。

## 6. 參考文獻


[1] Warren S. McCulloch and Walter H. Pitts. A logical calculus of the ideas immanent in nervous activity, Bulletin of Mathematical Biophysics 5, 115–133, (1943).

[2] Yann LeCun, Leon Bottou, Yoshua Bengio and Patrick Haffner, Gradient-Based Learning Applied to Document Recognition, (1998).

[3] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, (2012).

[4] Karen Simonyan, Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv:1409.1556, (2014).

[5] Min Lin, Qiang Chen, Shuicheng Yan, Network In Network, arXiv:1312.4400, (2014).

[6] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, Going deeper with convolutions, arXiv:1409.4842, (2014).

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. arXiv:1512.03385(2015)

[8] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger, Densely Connected Convolutional Networks, arXiv:1608.06993, (2015).

[9] Sergey Ioffe, Christian Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, arXiv:1502.03167, (2015).

[10] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton, Cifar10 and Cifar100 datasets, (2009).

[11] Sergey Ioffe, Batch Renormalization: Towards Reducing Minibatch Dependence in Batch-Normalized Models, arXiv:1702.03275, (2017).

[12] SchubertSlySchubert, Kaggle Cat and Dog, (2017).