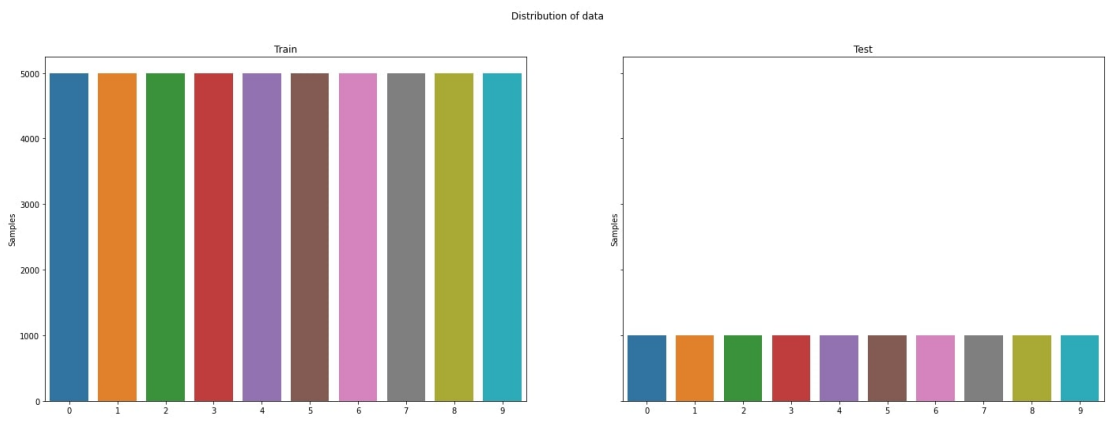


類神經網路簡介 HW5

Q36111281 張偉治

1. Data Processing

這次報告所使用的資料集為 CIFAR-10，是一組來自加拿大高級研究所的資料集，該資料集包含 10 種不同類別的 32*32 彩色圖片，其中原資料集將該資料集分為 50000 筆 train_data 以及 10000 筆 test_data。該資料集特點是解析度較小，能讓使用者快速驗證那些機器學習算法和神經網路模型是有效的，並且每個標籤數量都是分布平均的。同時為了在有限的計算資源下合理地分析三種不同模型的圖像辨識效果，我沒有對資料進行任何前處理。



2. Parameter Control and Architectural of AlexNet

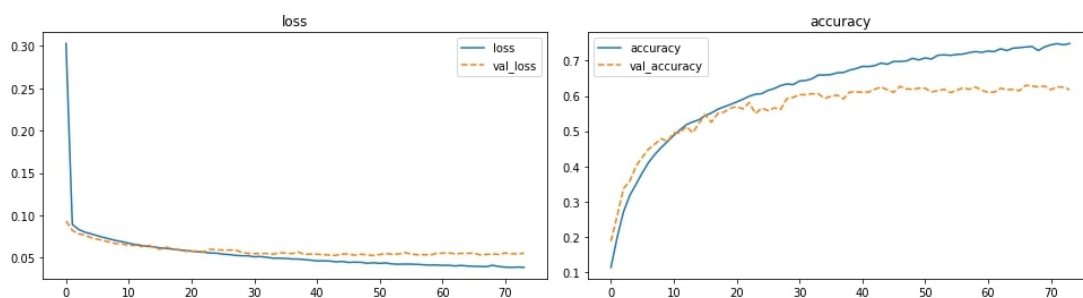
AlexNet 的基本想法是由五個卷積層與三個全連接層共八層組合而成，為了適應小解析度的圖像，我對經典的輸入(224,224,3)、輸出(1000)的 AlexNet 進行以下修正：

經典 AlexNet	修正後 AlexNet
Layer1：卷積層	
96 個 11×11×3 大小的卷積核掃描，步長為 4。使用 ReLU 激活函數。	12 個 3×3×3 大小的卷積核掃描，步長為 2。使用 ReLU 激活函數。
maxpooling 池化	
Layer2：卷積層	
256 個 5×5×96 的卷積核，步長為 1，做了 padding，padding 長度為 2。使用 ReLU 激活函數。	32 個 5×5×12 的卷積核，步長為 1，但是做了 padding，padding 長度為 2。使用 ReLU 激活函數。
maxpooling 池化	
Layer3：卷積層	
384 個 3×3×256 的卷積核，步長為 1，做了 padding，padding 長度為 1。	48 個 3×3×32 的卷積核，步長為 1，做了 padding，padding 長度為 1。使

使用 ReLU 激活函數。	用 ReLU 激活函數。
Layer4：卷積層	
384 個 $3 \times 3 \times 384$ 的卷積核，步長為 1，做了 padding，padding 長度為 1。使用 ReLU 激活函數。	48 個 $3 \times 3 \times 48$ 的卷積核，步長為 1，做了 padding，padding 長度為 1。使用 ReLU 激活函數。
Layer5：卷積層	
256 個 $3 \times 3 \times 384$ 的卷積核，步長為 1，做了 padding，padding 長度為 1。使用 ReLU 激活函數。	32 個 $5 \times 5 \times 48$ 的卷積核，步長為 1，但是做了 padding，padding 長度為 2。使用 ReLU 激活函數。
maxpooling 池化	
全連接層	
Dense=4096	Dense=128
全連接層	
Dense=4096	Dense=128

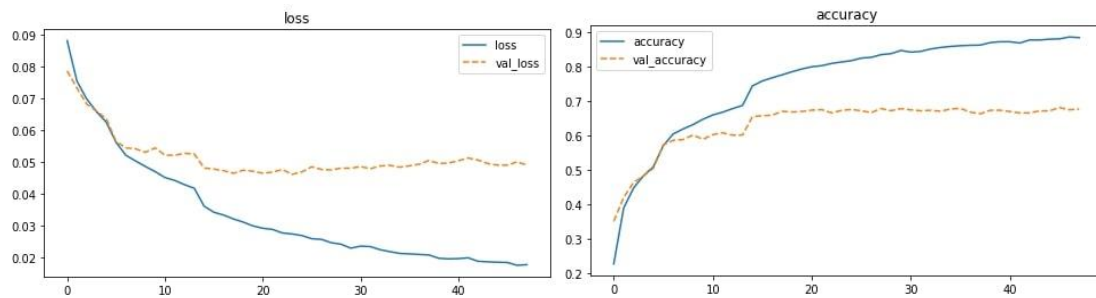
為了快速減少訓練時間即參數數量，我將每一層的卷積核數量都除以 8，將全連接層的神經數量除以 32，使用等比例的縮小能大程度地保留 AlexNet 的精神，希望卷積核的深度能隨著訓練層數增加越來越深。不過這邊有項參數設定的並不適切，就是第一層卷積層的卷積核大小，應該將大小設定為(7,7)或是(5,5)，這樣才能在卷積過程中獲得較多周邊像素的資訊。不過由於在撰寫報告時，colab 已經達到免費的 GPU 使用上限，因此無法實驗若修改第一層卷積層的卷積核大小是否有助於提升圖像辨識的精確度。

使用 CIFAR-10 訓練，訓練集精確度 0.74 並且有持續上升的趨勢，測試集精確度則穩定於 0.6 左右。



3. Parameter Control and Architectural of VGG

VGG 也是 Deep learning 中的一經典模型，其主要特點是與 AlexNet 不同，VGG 傾向選擇較小的卷積核，例如(3,3)的卷積核，減少參數的數量，並且選擇較深的網路。為了適應小解析度的圖像，我使用兩組連續兩層的卷積層，卷積核數量分為第一組 8 個大小為(3,3)卷積核、第二組 16 個大小為(3,3)卷積核，三組連續三層的捲積層，卷積核數量分為第一組 32 個大小為(3,3)卷積核、第二組 64 個大小為(3,3)卷積核、第三組也是 64 個大小為(3,3)卷積核。最後經過平坦化後進入兩層神經數量皆為 512 的全連接層。



與 AlexNet 相比，我所修改的 VGG 模型，多了兩個連續的卷積，並且增加了全連接層的神經數量，雖然增加了 6 倍多的參數量，不過同時在訓練集以及測試集的成績皆有大幅度的提升。使用 CIFAR-10 訓練，訓練集精確度從 0.74 提升至 0.88，測試集精確度則從 0.6 提升至 0.67。從結果可以發現 VGG 雖然相比 AlexNet 能更好的擬和訓練集並且收斂更快，但對於從訓練集中學習到如何判斷不同類別的特徵仍舊有所侷限。

4. Parameter Control and Architectural of ResNet

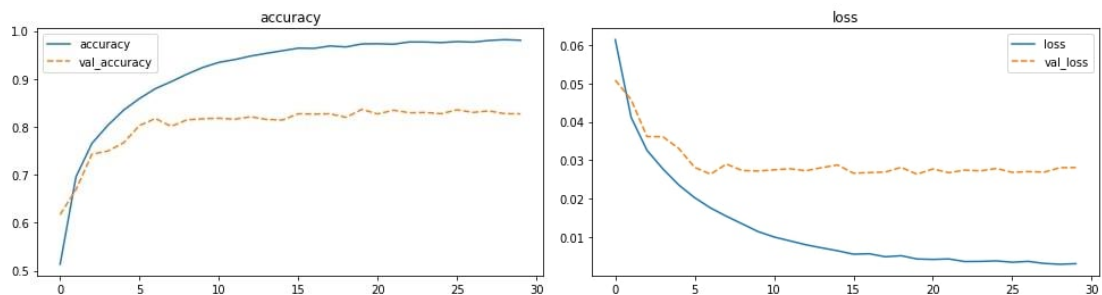
當卷積神經網路模型深度加深時，將導致兩個重要的問題，其一是特徵梯度消失，其二是模型參數量增加，ResNet 即是透過殘差的想法解決模型深度增加將導致的兩個問題，進而實現增加層數。

ResNet 將輸入複製成兩份，其中一份輸入卷積層後得到殘差，另一份不通過卷積層的稱為輸入，接著將輸入與殘差相加後得到下一層的新輸入。藉由此方法，若殘差值為零時，新輸入等於輸入，此時為恆等映射，因此若這一層沒有學到新的特徵，也不會讓模型退化。

我所設計的殘差網路架構如下表，我所設計的 ResNet 卷積層數為 19，相比 VGG 模型多了 6 層的卷積層，同時增加了卷積層的通道數量，但相比 VGG，參數數量僅增加 1.8 倍。值得注意的是，在不同 Block 轉換時，因為輸入與殘差大小不同，因此輸入需要額外做一次卷積將大小與殘差之間的大小正規化。

Block	Layer name	Output size	Kernel size	Repeat times
BN		32*32	[32,32,3]	
Conv_32		32*32	[32,32,32]	
Relu+BN		32*32	[32,32,32]	
1	Conv_32	32*32	[32,32,32]	2
	Relu+BN	32*32		
	Conv_32	32*32		
	Add	32*32		
	Relu+BN	32*32		
2	Conv_64	16*16	[16,16,64]	5
	Relu+BN	16*16		
	Conv_64	16*16		
	Add	16*16		
	Relu+BN	16*16		
3	Conv_128	8*8	[8,8,128]	2
	Relu+BN	8*8		
	Conv_128	8*8		
	Add	8*8		
	Relu+BN	8*8		
Pooling		2*2	[2,2,128]	
Flatten		512	[512]	
Dence_Relu		64	[64]	
Dence_SoftMax		10	[10]	

與 VGG 相比，在參數量僅增加了 1.8 倍多的情況下，使用 CIFAR-10 訓練，訓練集精確度從 0.88 提升至 0.98，測試集精確度則從 0.67 提升至 0.82。可以發現 ResNet 架構對於訓練集的擬和效果是三者之中最好的，同時不同於 VGG 之於 AlexNet，ResNet 的測試集精確度有大幅度地提升。

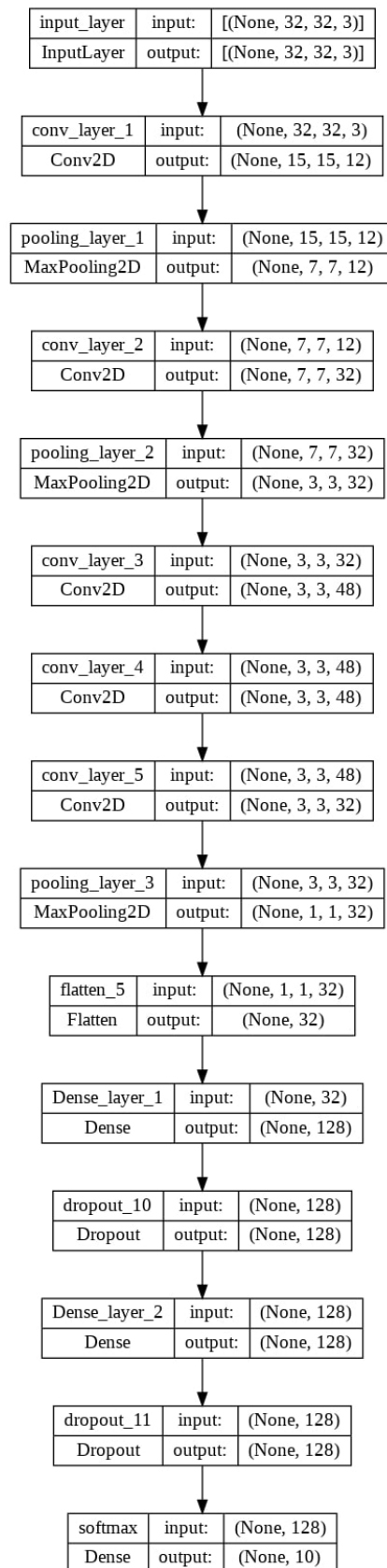


5. Analysis and Comparison of Three Models

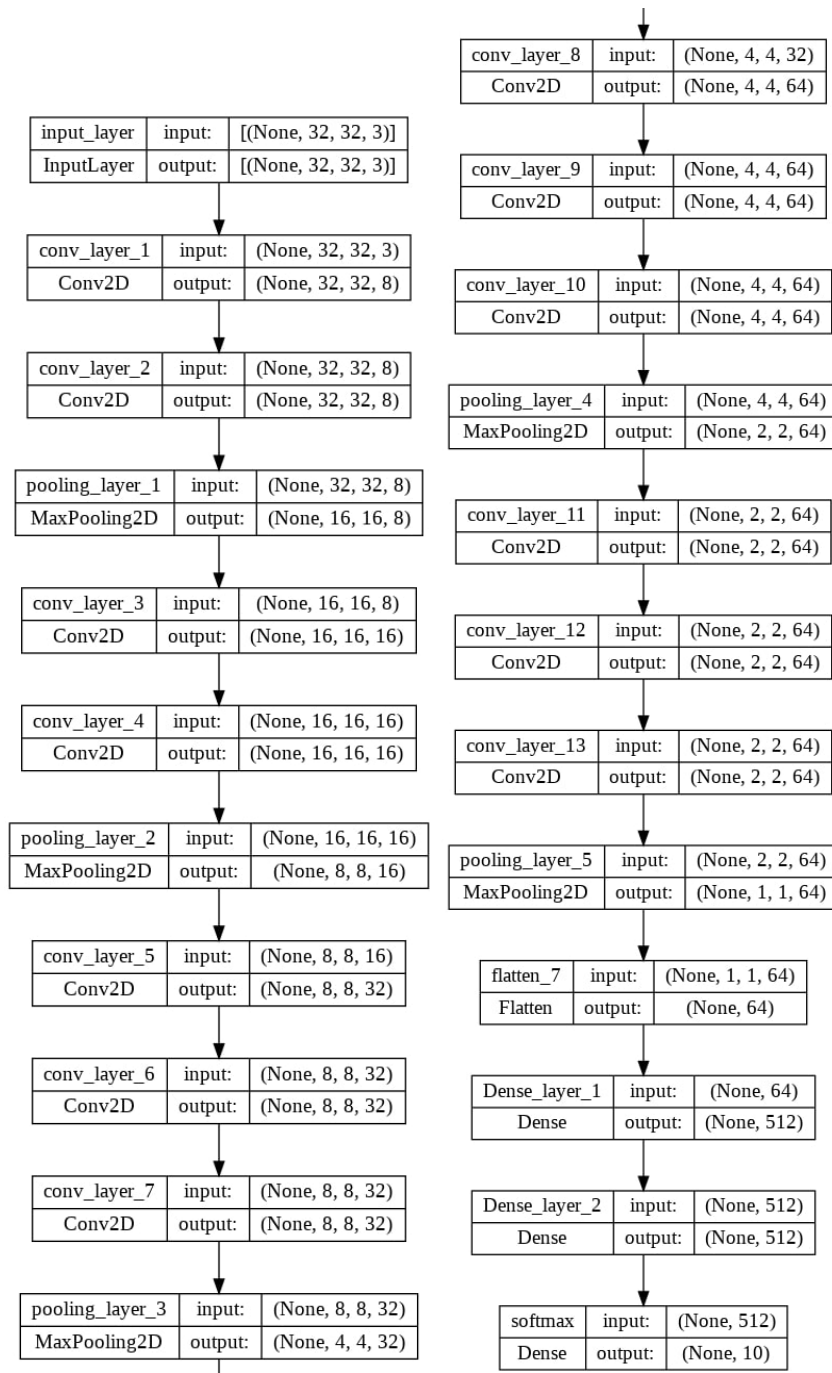
三種模型在 Kaggle 上的成績趨勢與使用 CIFAR-10 資料集測試驗證時的趨勢相同，不過在資料集時成績最好的 ResNet，在 Kaggle 上準確率僅為 0.5，可以推測對於未知的圖像，是否進行資料的前處理可能導致在辨識未知圖像時準確率有高達 1.6 倍的差異($0.5 \rightarrow 0.8$)。透過隨機選擇 12 張圖片並標示出模型的辨識結果，與人工辨識的結果相比較，準確率約為 0.4，同時觀察辨識的結果，可以推論模型辨識圖像類別的大方向，是透過物體的輪廓進行推論，例如馬的臉呈現く字形時，模型將其辨識為鳥類，當影像是正在奔跑的馬時，模型就正確預測其類別。



附錄 1：AlexNet architectural



附錄 2：VGG architectural



附錄 3：ResNet architectural

