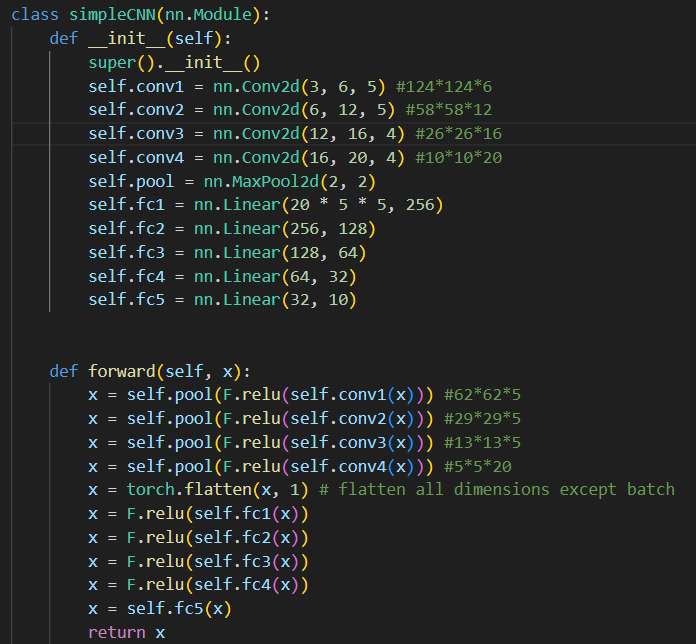
HW4: CNN

109511219 林錦樑

整個classifier的架構參考[PyTorch官方教學](https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html)

1. CNN
   1. Baseline

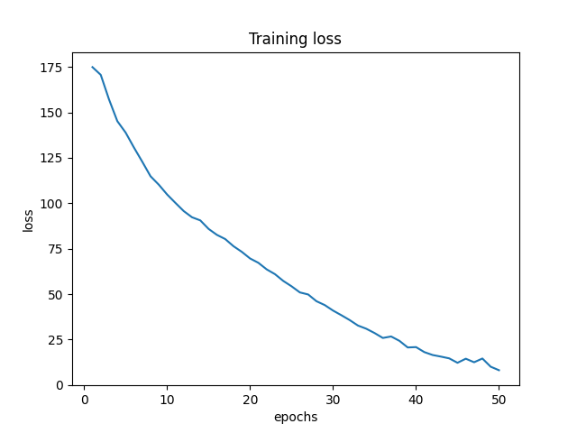
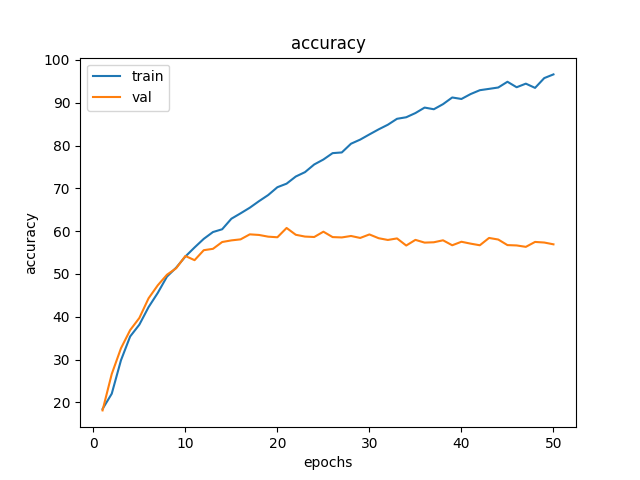
Baseline的CNN model我使用了四層convolution與max pooling，萃取出特徵後再使用五層FC layer，其結構如下圖。



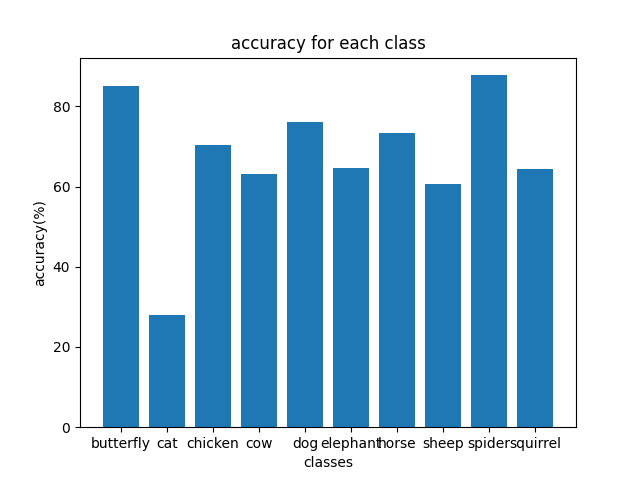
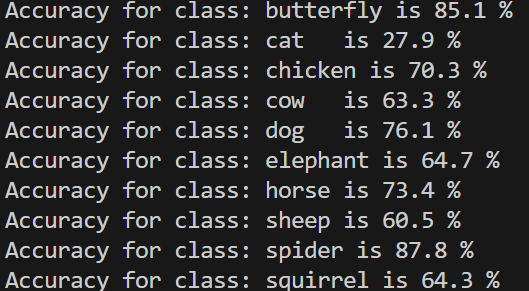
Hyperparameters的選擇上包括batch size為256，optimizer使用Adam，learning rate設為0.001，總共訓練50個epochs。資料的部分圖片輸入大小為128\*128，各種類圖片資料量如下表格。

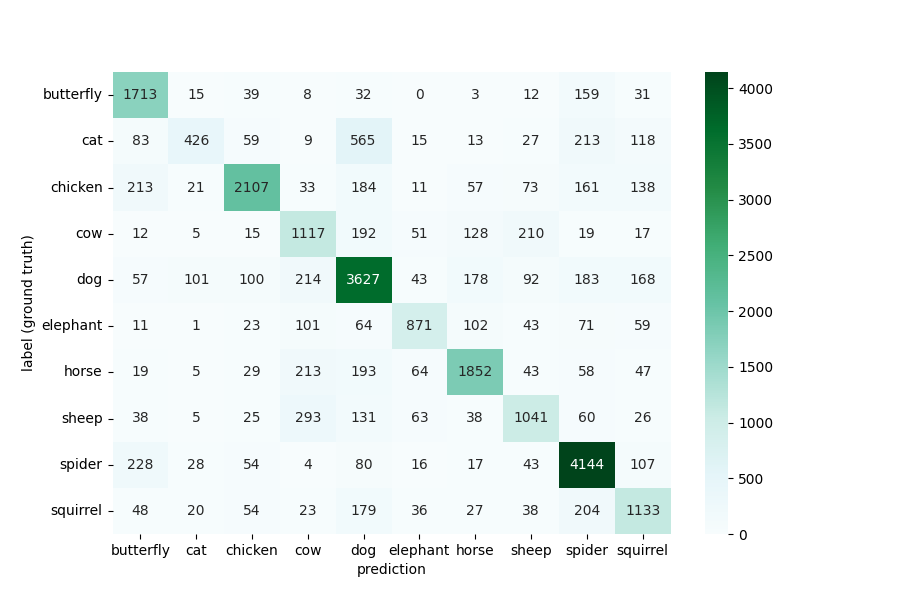
|  |  |
| --- | --- |
| 類別 | 數量 |
| butterfly | 2012 |
| cat | 1528 |
| chicken | 2998 |
| cow | 1766 |
| dog | 4763 |
| elephant | 1346 |
| horse | 2523 |
| sheep | 1720 |
| spider | 4721 |
| squirrel | 1762 |

經過50個epochs後，其accuracy與loss趨勢如下圖。可以看出loss隨epoch增加持續下降，accuracy在大概第15個epoch後，training accuracy能持續上升，而validation accuracy在55~60%震盪，可以推測訓練過程出現overfitting的現象。



從全部資料得出的各類別的accuracy與confusion matrix來看，cat的表現最差，常常誤分到dog，其可能原因除了兩者相似之外，cat的訓練資料最少也會導致model對cat的分類能力較差。而其他訓練圖片少於兩千張的類別，其準確率皆低於70%；訓練圖片大於四千張的類別，其準確率可大於85%，可見訓練圖片數量會影響模型分類表現。

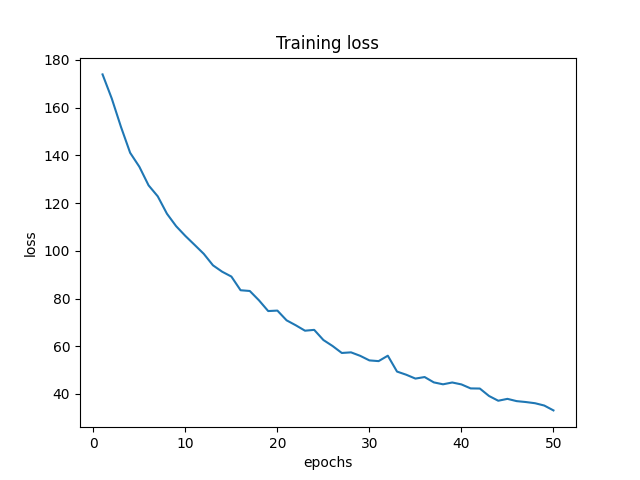
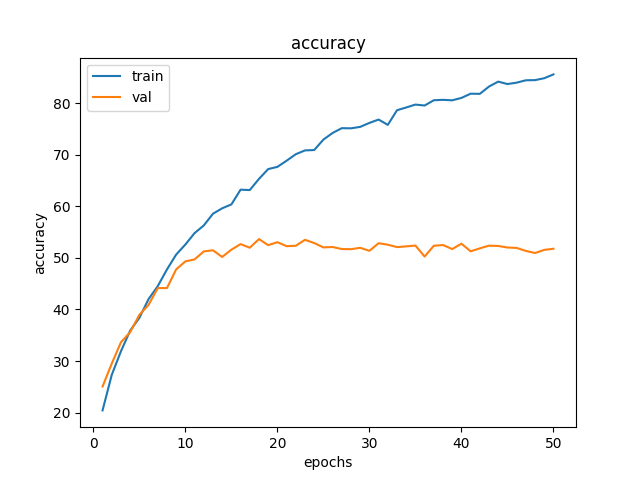




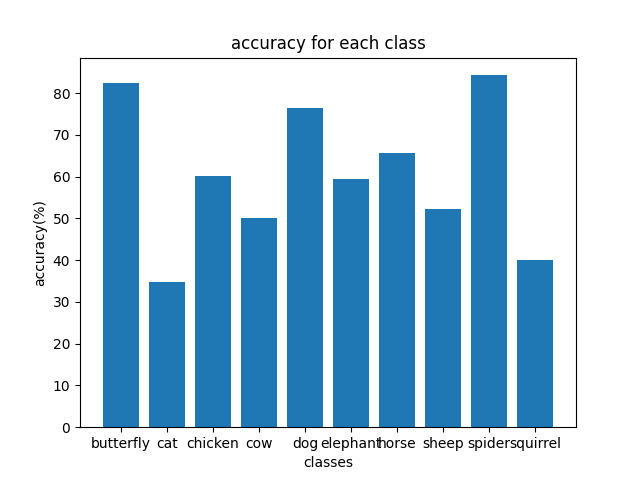
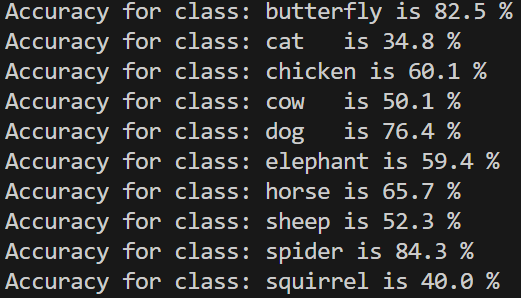
* 1. L2 Regularization

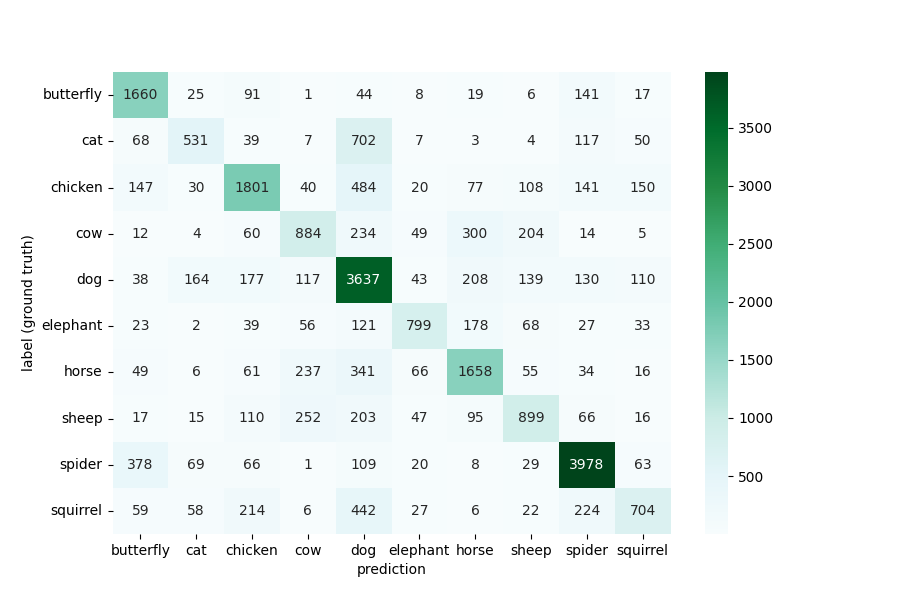
由於出現overfitting的現象，我先利用L2 Regularization的方式限制參數大小，試著減緩overfitting的情形。在optimizer的參數上加上1e-5的weight\_decay，並將learning rate調為0.005，訓練結果如下圖。

但其結果並不盡理想，不論是training或是validation的accuracy都降低了約10%左右，loss的部分則是由於加入了Regularization term，有出現增加的情況。



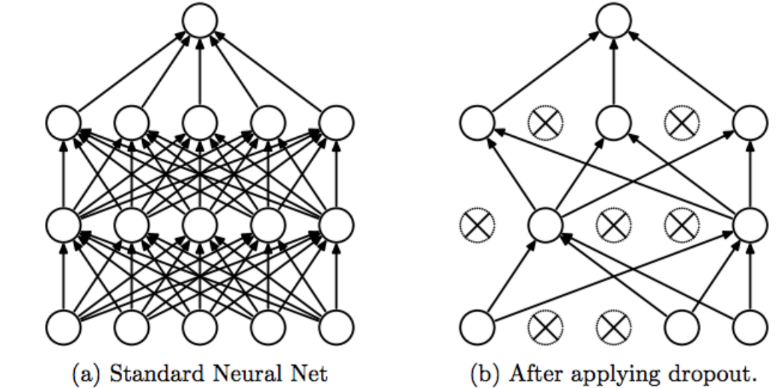
從全部資料得出的各類別的accuracy與confusion matrix來看，cat的表現有變好，但誤分到dog的情況有增加，反倒是分類到其他類別的情況有減少。其他類別的準確率大部分下降，尤其以squirrel下降最多，最容易誤分到dog。從以上實驗可看出L2 Regularization的方法無法提升模型準確度。



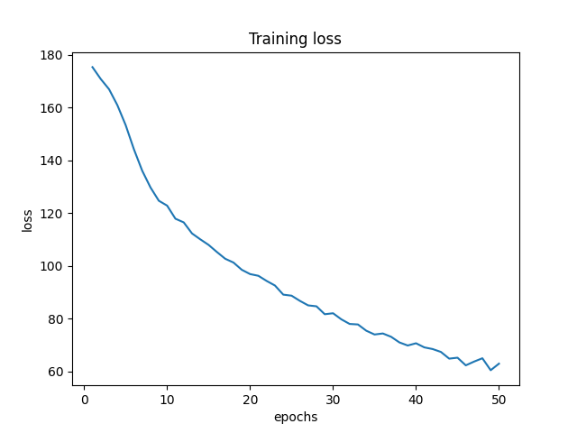
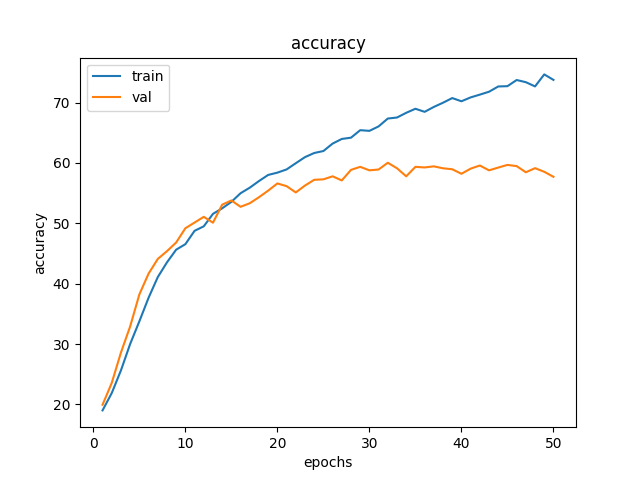


* 1. Dropout

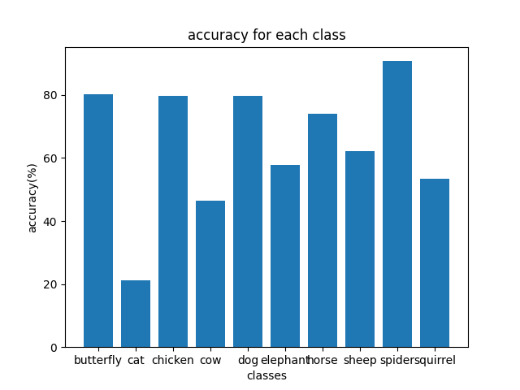
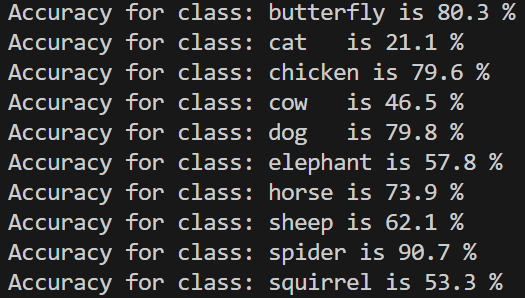
由於L2 Regularization無法解緩overfitting的問題，甚至帶來更差的結果，我決定改用dropout的方式來處理問題。Dropout會在前向傳播時，依照設定的機率關閉神經元，降低對於特定神經元的依賴，如下圖所示。在實作上我在每層Fully Connected Layer都加上了機率為0.3的dropout。

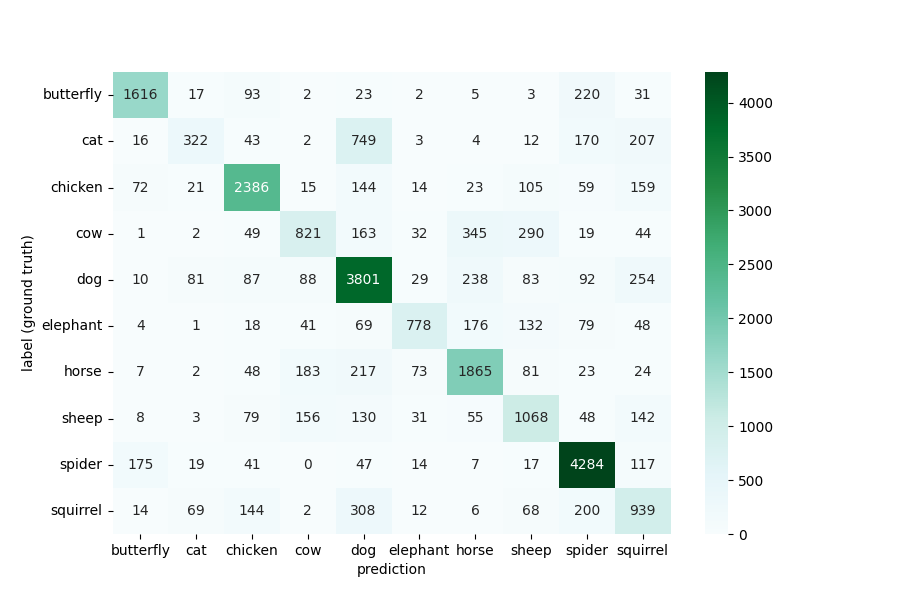
[](https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5)

加入dropout後的實驗結果如下，validation的accuracy回到跟baseline差不多水準，但training accuracy有下降，loss也有上升，可能需要經過更多epoch的訓練才能得出較baseline好的model。



從全部資料得出的各類別的accuracy與confusion matrix來看，進步與退步的類別各占一半。其中將cat誤分到dog的狀況變得更加嚴重。



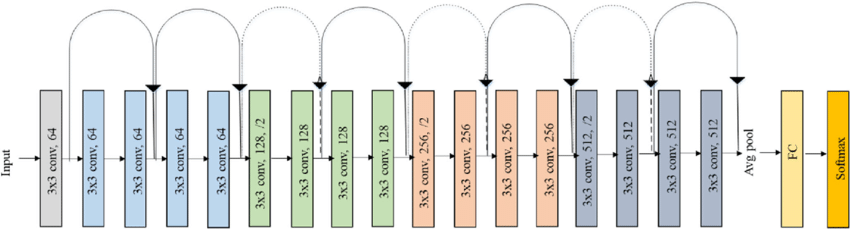


從以上結果來看，L2 Regularization與dropout都無法解決overfitting的問題，甚至會帶來更差的結果。我推測可能的原因來自訓練圖片不夠多，導致一些處理overfitting的方法不夠有效。也有可能嘗試改變這些方法的hyperparameter後，能帶來更好的效果。

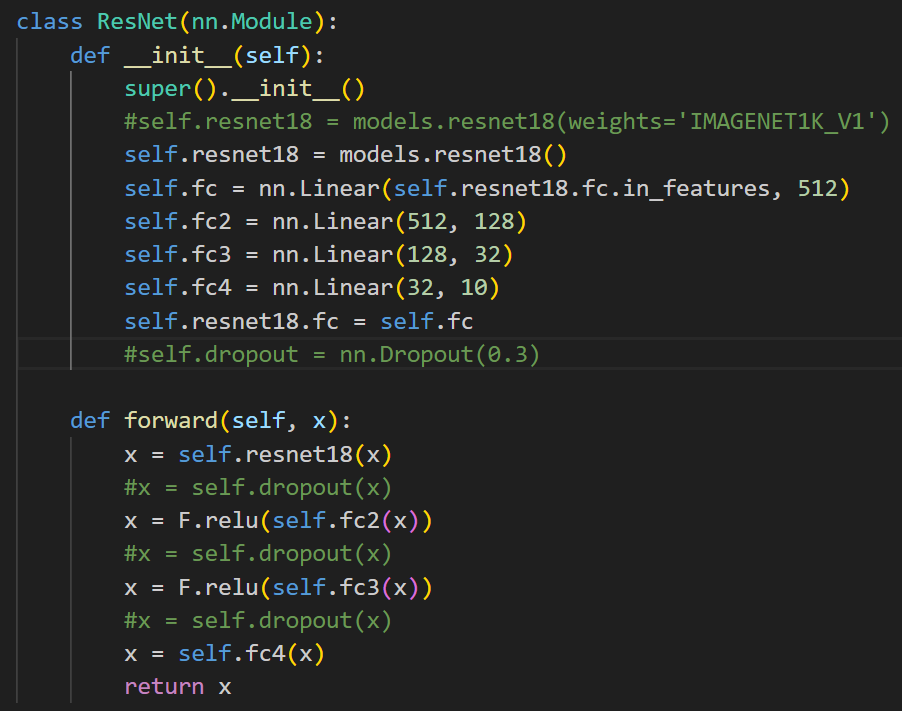
1. ResNet18
   1. Baseline

由於可能因訓練資料不足，而造成前面使用處理overfitting的方法來提升validation accuracy的效果都不盡理想，我嘗試使用一個更大的模型來同時提升training和validation accuracy，這邊以ResNet18來進行實驗。

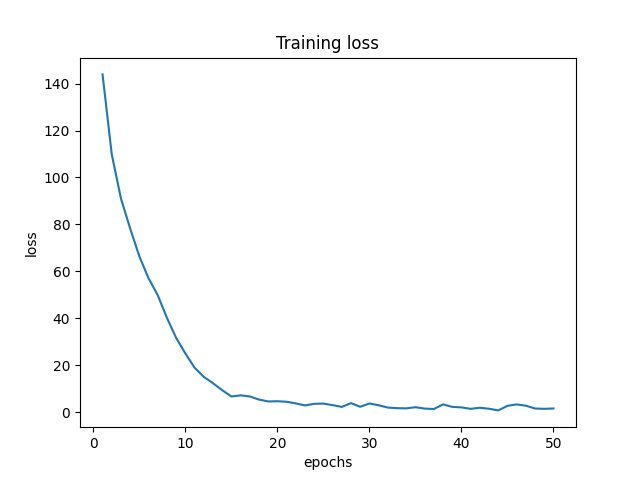
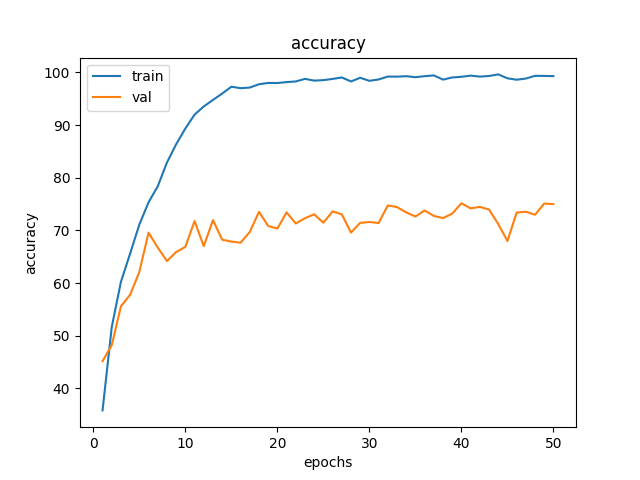
ResNet18包含17層的convolutional layer與1層的FC layer，共18層，其中由兩層convolutional layer所形成的residual block可以透過增加一條加法路線的方式，解決梯度消失的問題。整體架構如下圖，其中上方連線部分即為residual block的加法路線。

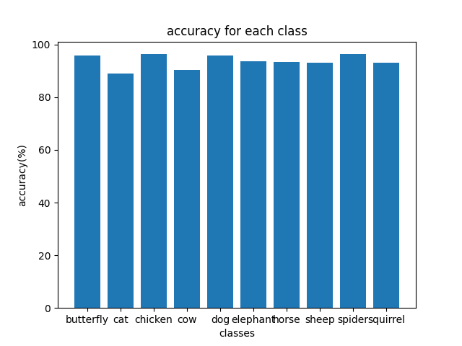
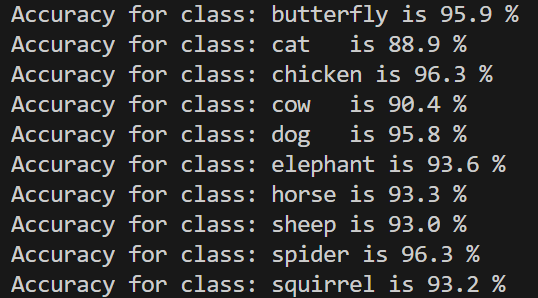
[](https://www.researchgate.net/figure/Original-ResNet-18-Architecture_fig1_336642248)

整個model的部分是經過一個ResNet18萃取出特徵後，再使用四層FC layer進行分類任務，其結構如下圖。



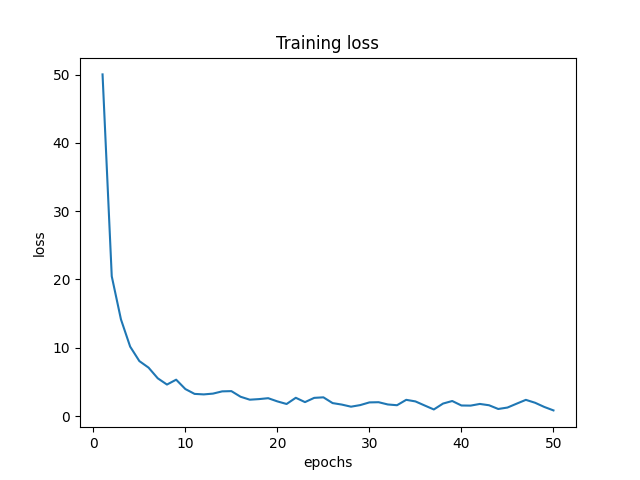
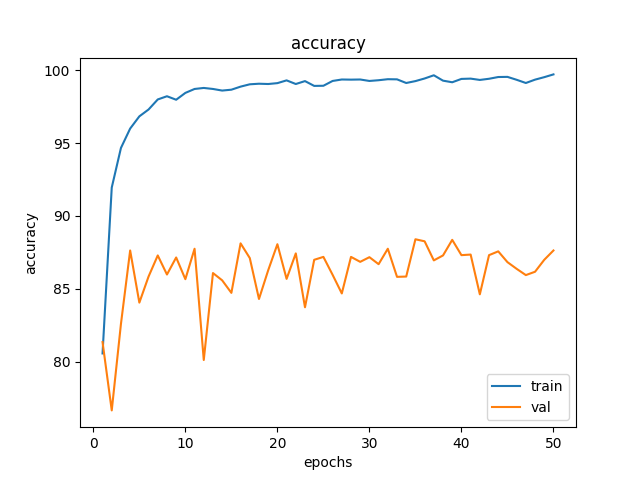
從整體accuracy來看，training accuracy比一般簡單的CNN上升速度還快，validation accuracy也增加了約10%左右。Loss的收斂速度也增加許多。從個別的accuracy來看，整體表現大幅提升，各類別accuracy都有大於88%，且各類別表現較為平均。可以看出使用更大的模型能帶來更好的效果，但仍舊存在overfitting的問題。

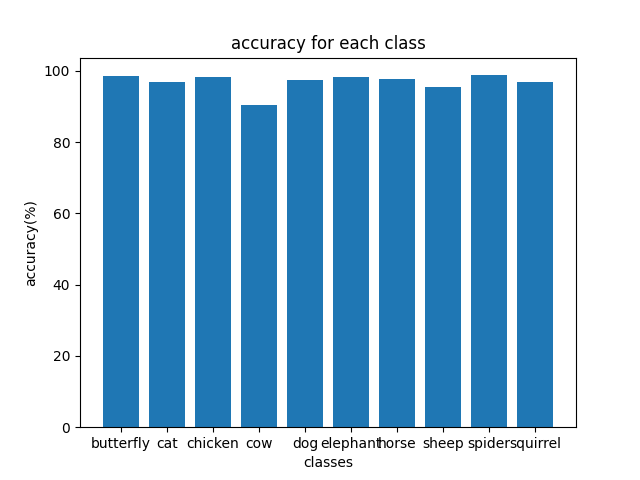
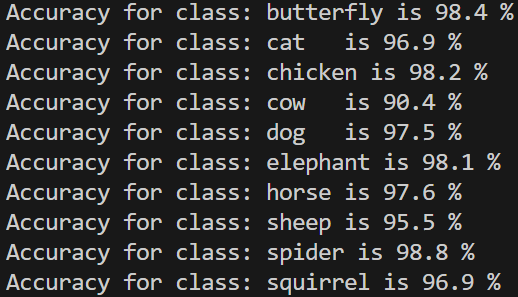


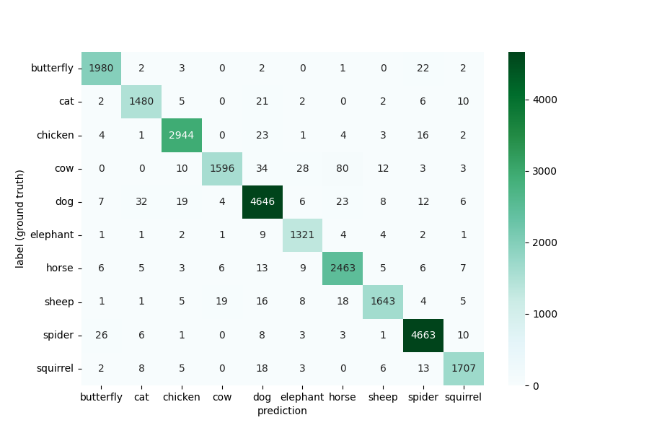




Pretrained model的部分是使用了IMAGENET1K\_V1這個weight，從整體與個別的accuracy來看皆有提升，validation的accuracy增加了10%～15%，各類別的accuracy均能大於90%，可以看出使用pretrained model能對這個分類任務帶來更好的效果。

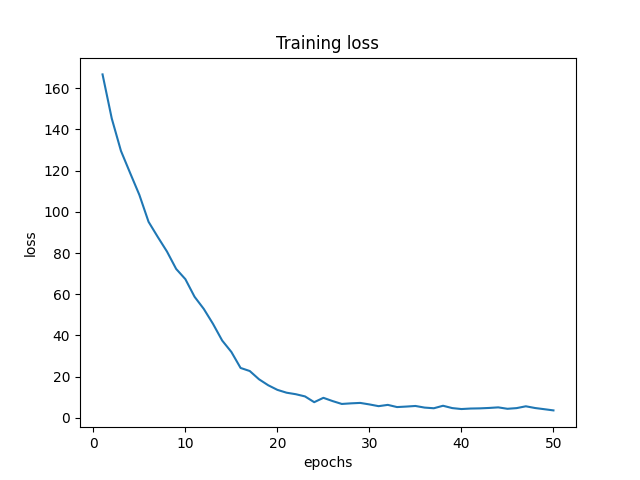
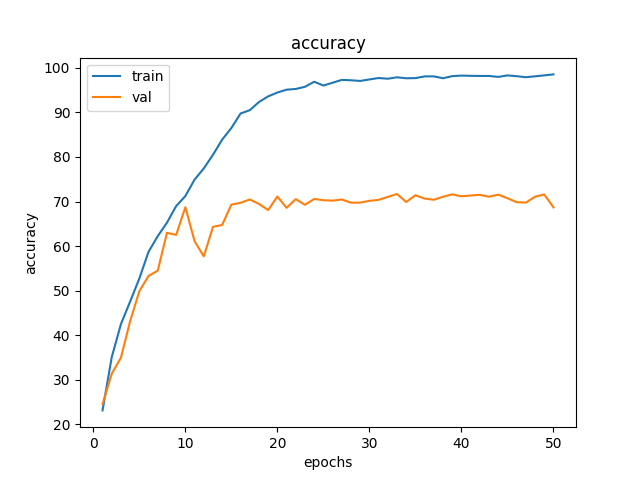


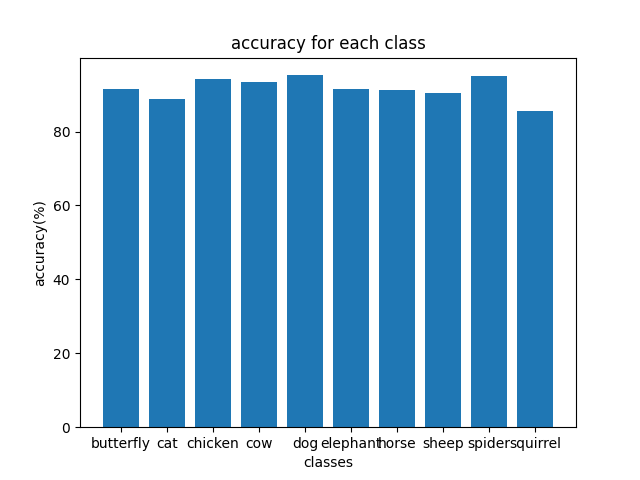
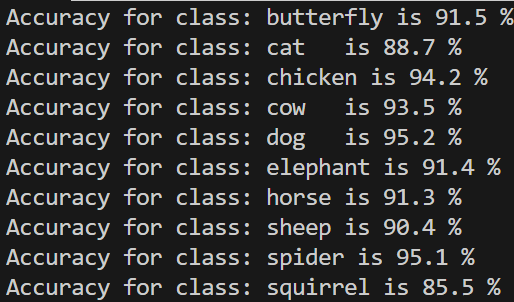




* 1. L2 Regularization

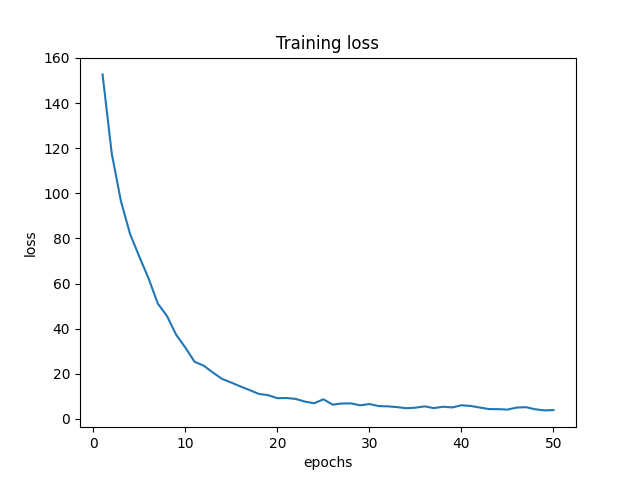
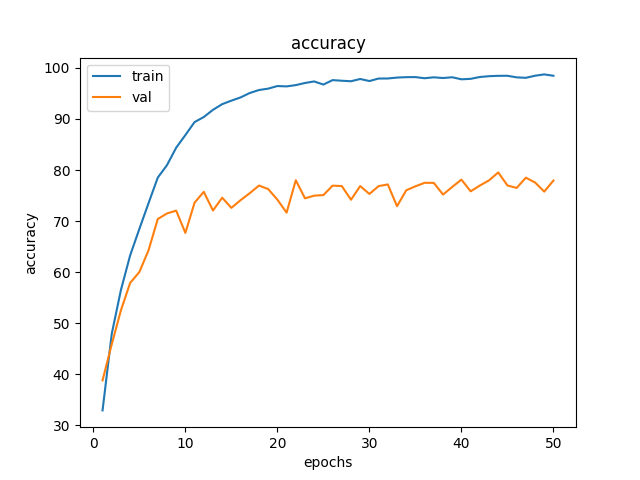
以下為使用L2 Regularization後的ResNet18 model的表現。從accuracy曲線、各類別accuracy與confusion matrix來看，皆無太大變化。

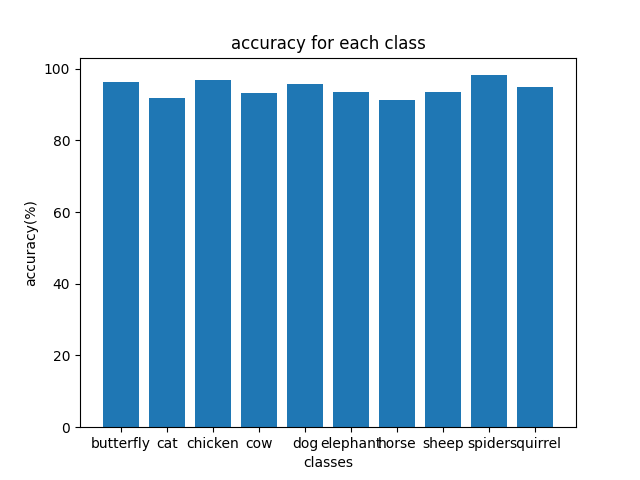
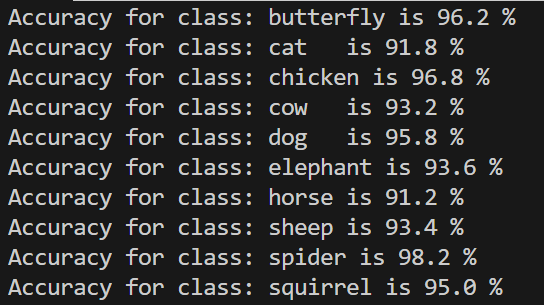


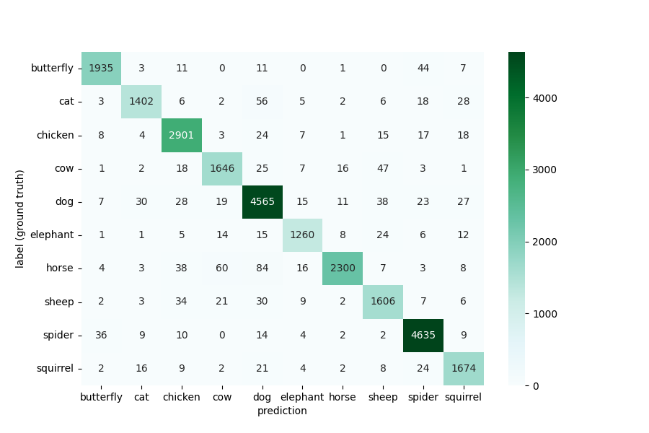




以下為使用L2 Regularization後的pretrained ResNet18 model的表現。從accuracy曲線來看，validation accuracy掉了約10%左右。從各類別accuracy與confusion matrix來看，大部分類別的accuracy皆下降。

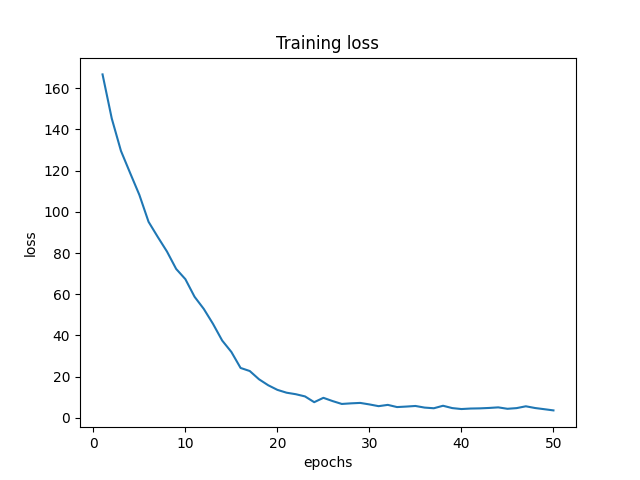
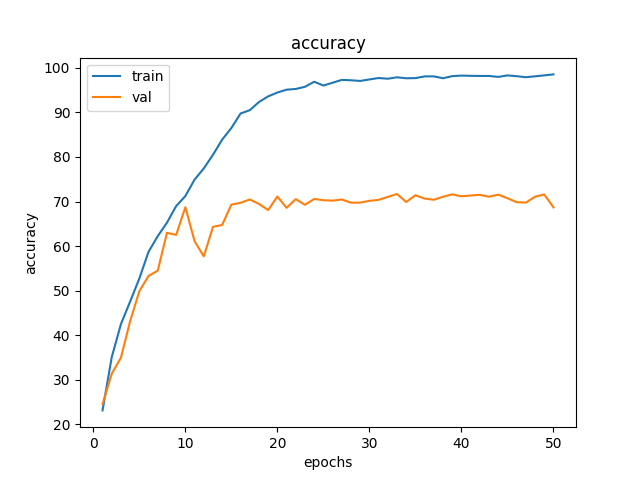


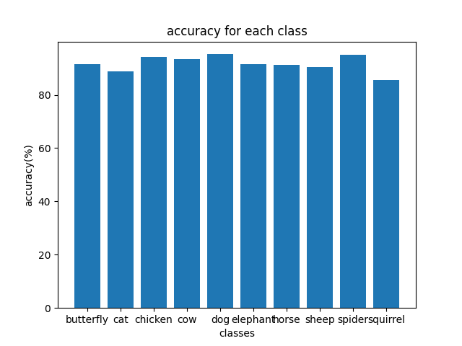
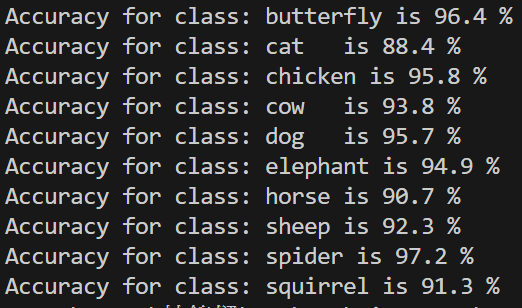




* 1. Dropout

以下為使用Dropout後的ResNet18 model的表現。從accuracy曲線、各類別accuracy與confusion matrix來看，皆無太大變化。







以下為使用Dropout後的pretrained ResNet18 model的表現。從accuracy曲線來看，validation accuracy掉了約10%左右。從各類別accuracy與confusion matrix來看，無太大變化。

