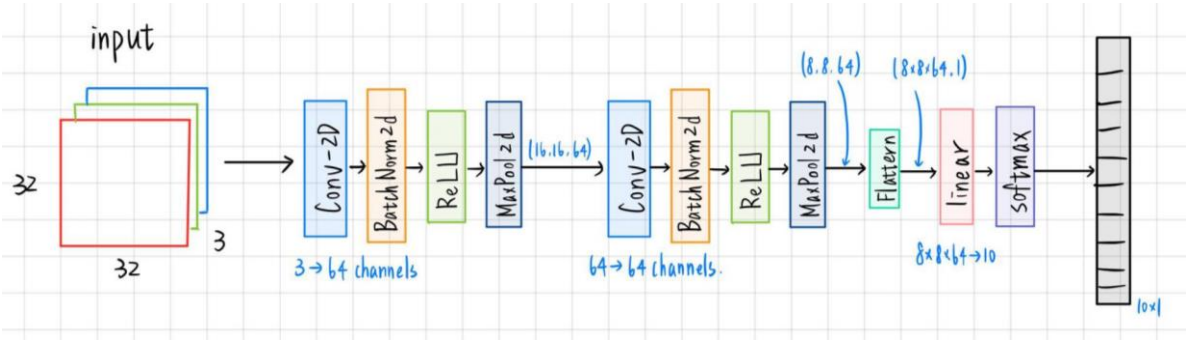
HW2 AI_Principle

110511068 蔡雅婷 110511164 張語楹 109261007 李紹穎

Code: https://github.com/Benchangatrul284/AI_principle/tree/main/HW2

Part1



part1，主要是搭建出 dataset 和 dataloader、訓練流程、test accuracy 的判斷，並使用一個小模型進行測驗。模型具有三層 hidden layers，這三層 hidden layers 是兩層 convolution 2d 和一層的 linear 作為分類器，是個非常輕量的模型。

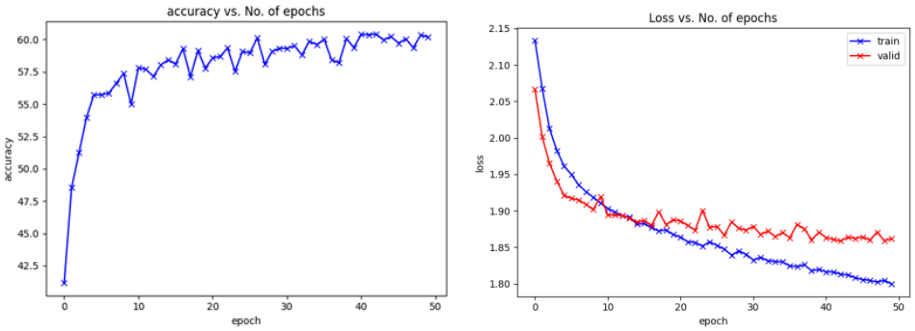
因為這次是處理圖片的 classification，圖片是二維的，使用 convolution2D 相比對資料 flatten 後過 linear 層具有可以減少參數量、並可保存圖片 pixel 和 pixel 間位置信息的特性，此外，增加 channel 數有助於增加模型的複雜度，讓模型每一個 channel 去學習在圖片中的不同特徵等等。

接著加上老師上課時教過的 batchNorm、ReLU、MaxPool。BatchNorm 有助於避免梯度消失、讓訓練更加穩定，activation function 是 ReLU，ReLU 實現非常簡單且有助於避免 gradient saturation，MaxPool 有助於讓訓練參數減少的同時，可以讓模型學習到更不同的特徵和抗雜訊功能等等，後兩者都是不會增加訓練參數的，而 batch normalization 的參數和 convolution 層或 linear 層相比微乎其微。

最終學習結果:

epochs = 50, batch size = 1024, learning rate = 5e-3

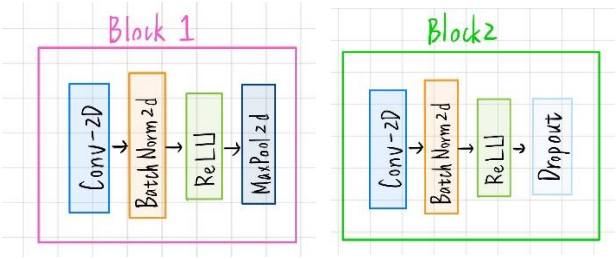
best accuracy =60.42%

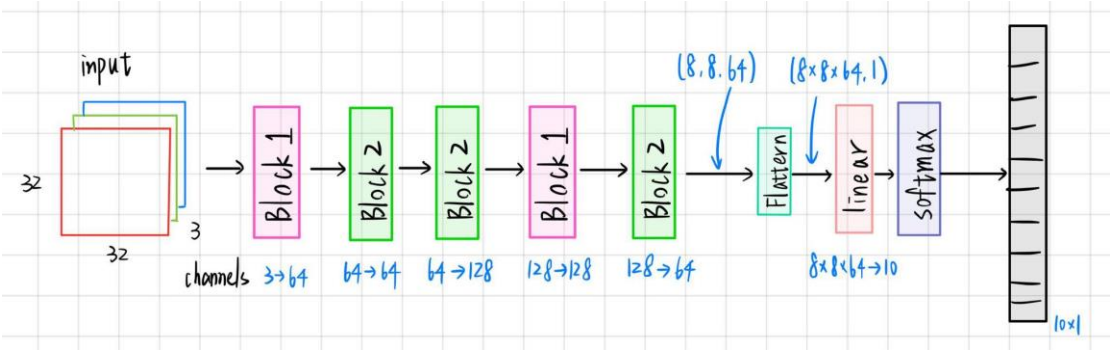


可以從中看出在大約 20 epoch 後，accuracy 就沒有顯著增加，validation loss 也沒有下降，因為模型較小無法學習更複雜的特徵。

Part2

這部分可以自行定義模型，為了讓模型能夠 accuracy 變好，我們選擇使用夠深更複雜的模型，參數量變多，也可以學習到圖片中更多的特徵，實作後發現準確率提高非常多。



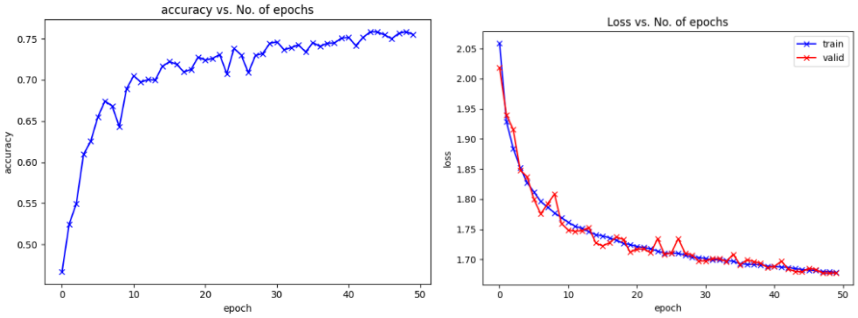


我們將原先三層的 hidden layer 增加到 6 層，分別是五層的 convolution 和一層的 linear 層，將 conv2d、batchnorm、ReLU、MaxPool、Dropout 分別形成 block1 和 block2，block1 是負責縮圖，讓參數量變小的同時還有抗雜訊的效果，block2 可以增加 model 層數和具有 dropout，可以減少過擬和。

最終學習結果:

epochs = 50, batch size = 1024, learning rate = 5e-3(維持)

best accuracy = 75.86%



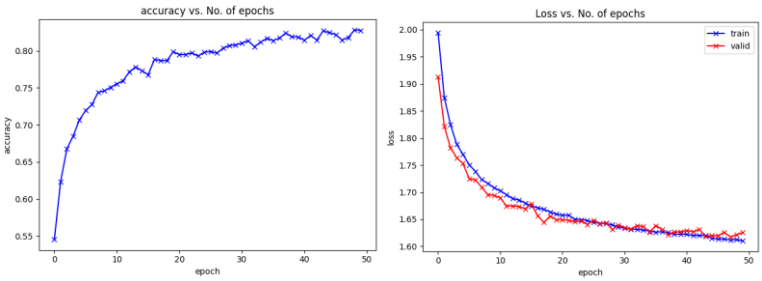
可以從結果看出增加深度確實有助於模型獲得更高 accuracy，且達到 75.86%的成績。

Part 3

這部分要調超參數 hyperparameter。我們會從 batch size, data augmentation, optimizer, 調整模型架構，參數量的角度出發。

Batch size: 減少 batch size 為 128:

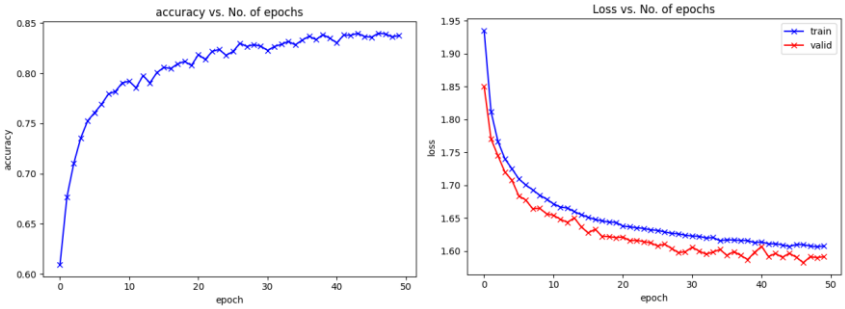
Best Accuracy: 82.83%



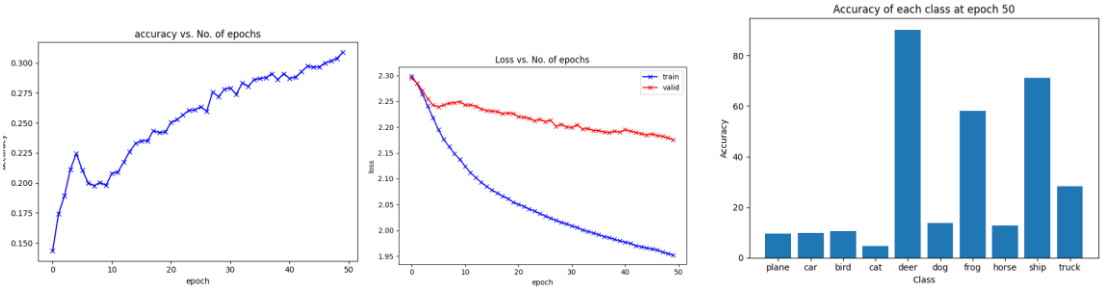
此後實驗都會使用此 batch size。

增加 data augmentation: 為了增加訓練資料，我們將資料做水平翻轉來增加訓練資料的變異度，防止 model 單純背下 training data。

Best Accuracy: 83.98%，可以看到 accuracy 略微提升。



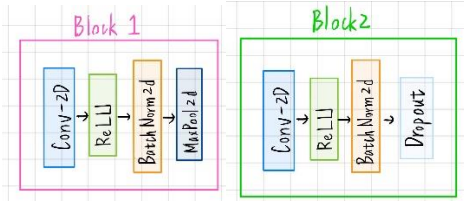
使用 SGD 來當作 optimizer:



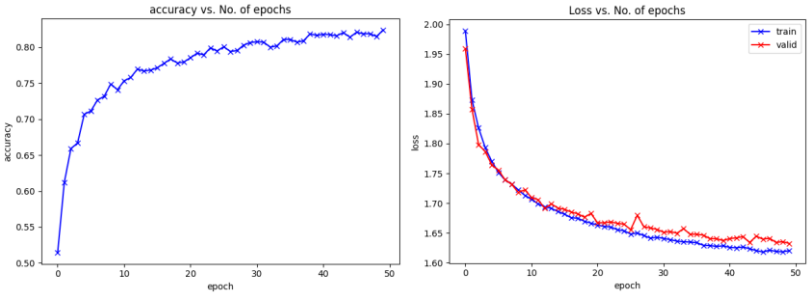
SGD 的效果不如 Adam(Adaptive Moment Estimation)好，除此之外，可以看到很每個 class 的 accuracy 相差甚遠。

更改 model 的架構:

將 activation function 移到 batch normalization 前面。



Best Accuracy: 82.41%

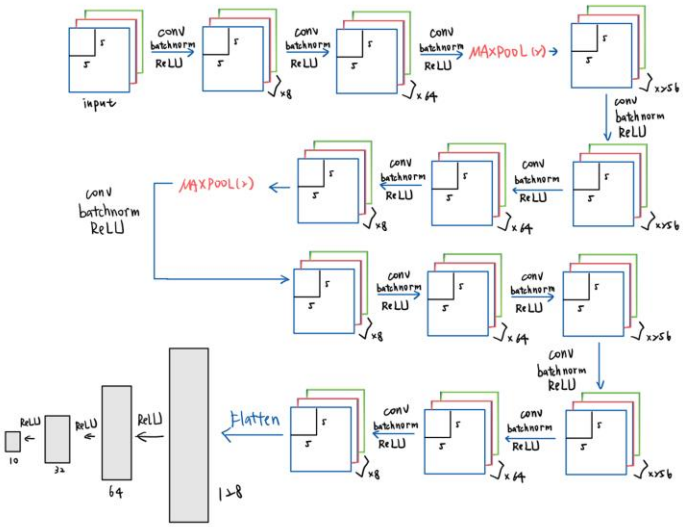


就結果而論，略遜於調換前的 82.83%。

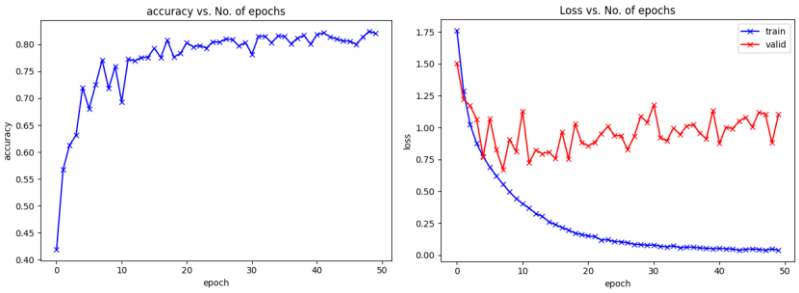
調整深度及參數量:

Params: from 0.38M -> 4.98M Flops: from 0.14G -> 2.04G

模型如下圖，共有 16 層可訓練的參數:



Best Accuracy: 82.43%

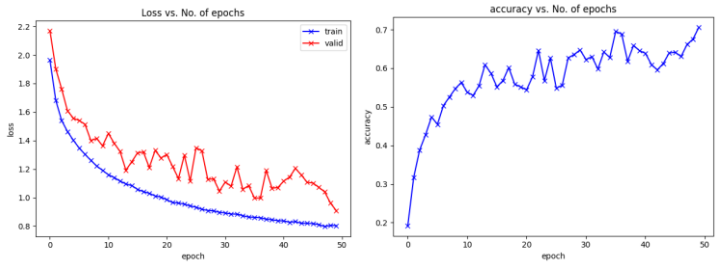


Accuracy 意外的下降，可以看到 validation loss 雖然和前幾個實驗相比較小，但和 training loss 相比大的很多，因此這是很明顯的 overfitting。

Dropout and weight decay:

當層數越來越多時，可以發現效果並沒有想像中的那麼好。首先，越深的 model 代表參數越多，模型的擬和能力越好，越容易 overfit。

因此，我們將 model 做更多的 regularization，如增加更多的 Dropout 和在 Adam 中增加 weight decay。



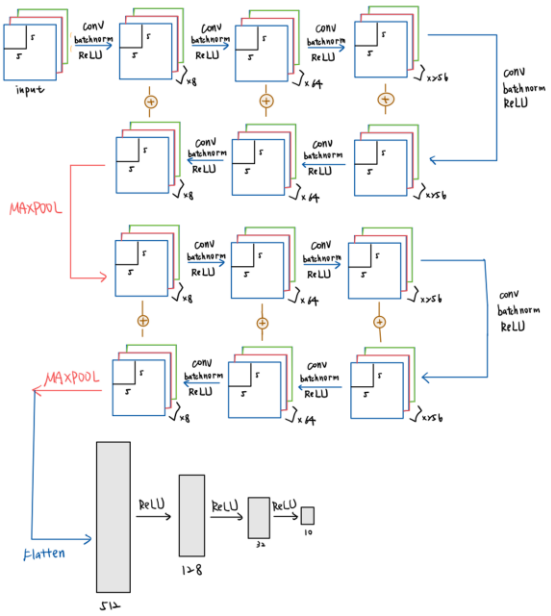
Best Accuracy: 70.63%

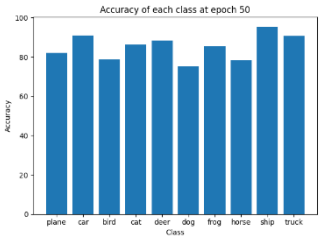
Overfitting 的效應確實減少許多。但 accuracy 卻也減少了。

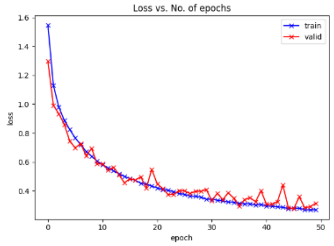
Part 4 Further study of CNN:

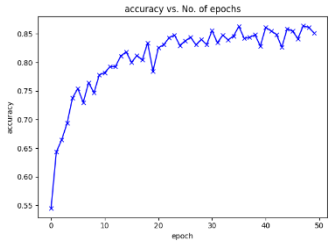
此外，因為更新 weight 適用 back propagation 不斷累乘，因此在前面幾個 layer 可能會遇到 gradient vanishing 的問題。針對這個問題，在 2015 有人就提出使用 residual link 來解決這個問題。Residual link 就是使用 identity map 直接加到後面的幾層，因此在 back propagation 中，能夠跳過中間幾項的累乘達到穩定訓練的效果。在本題就是加上棕色的連結。

Best Accuracy: 0.8643









Conclusion:

這次作業完成了機器學習最常見的分類問題，也可以看到 hyperparameter 對於訓練結果的重要性。如何決定 hyperparameter 呢?我們認為要根據該問題的 domain knowledge。此外，我們也發現模型並不是越大越好，而是要根據問題來選擇對應的模型。如在 part1 所提及，CNN 在圖像的問題上就比 linear 來的適合。