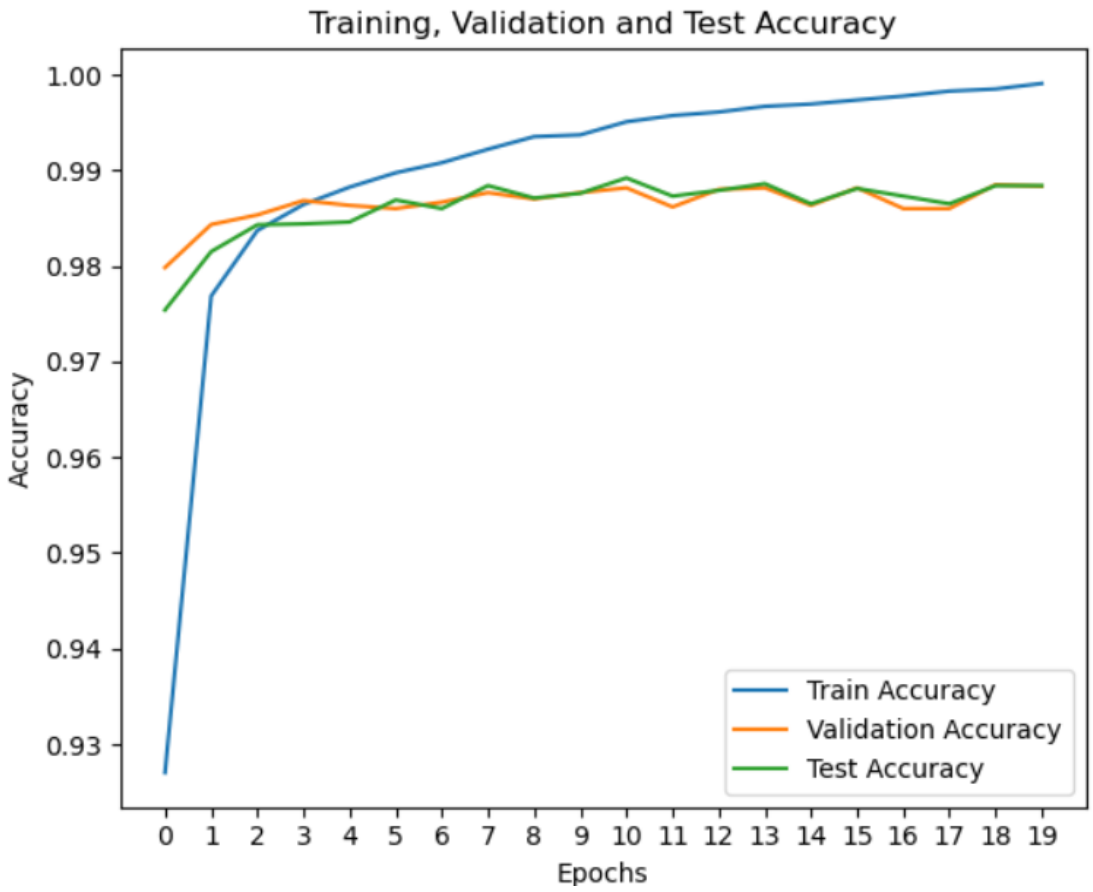
Part I.MNIST

Convolutional Neural Network Structure:

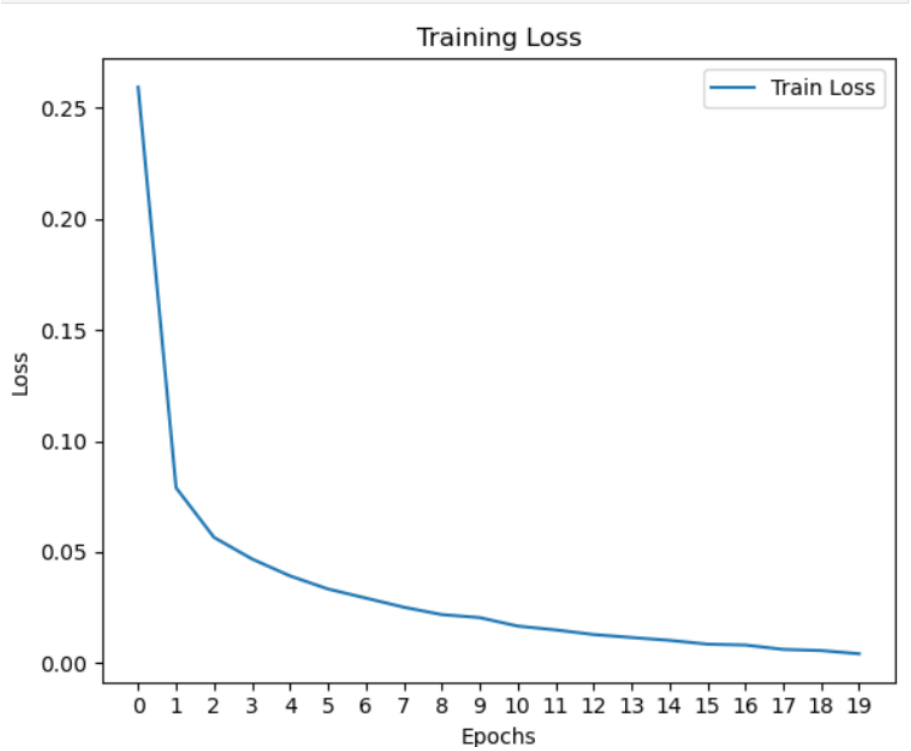
Filter size:(5, 5), strides:(1,1)

	Layer Name	Layer Type	Output Shape
0	input_layer_2	InputLayer	(None, 28, 28, 1)
1	conv1	Conv2D	(None, 24, 24, 64)
2	pool1	MaxPooling2D	(None, 12, 12, 64)
3	flatten_2	Flatten	(None, 9216)
4	dense_2	Dense	(None, 10)

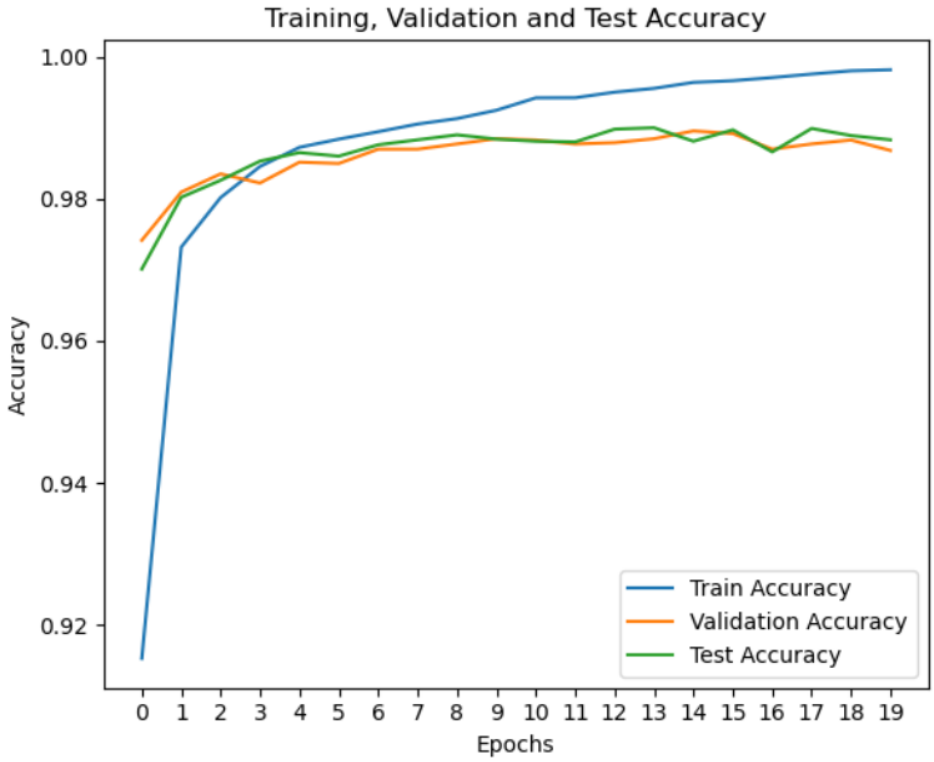
Accuracy plot

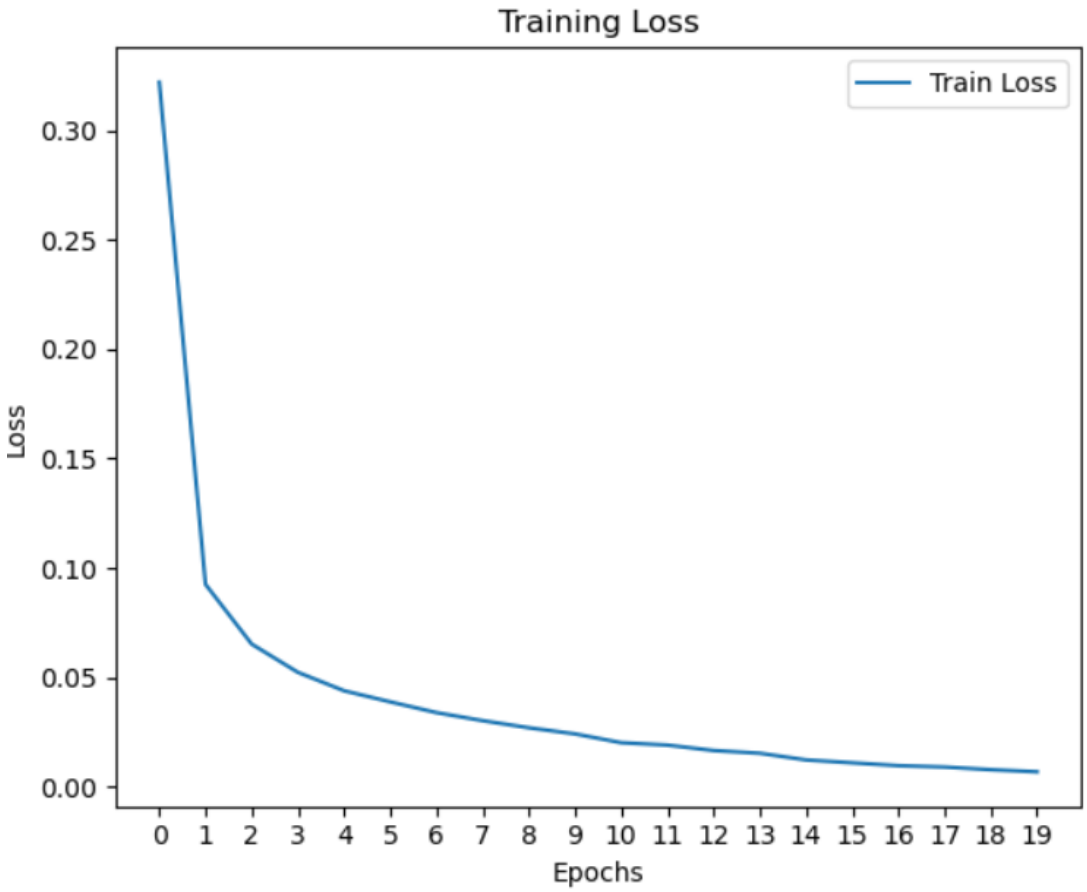


Loss plot



Filter size:(10,10), strides:(2,2)





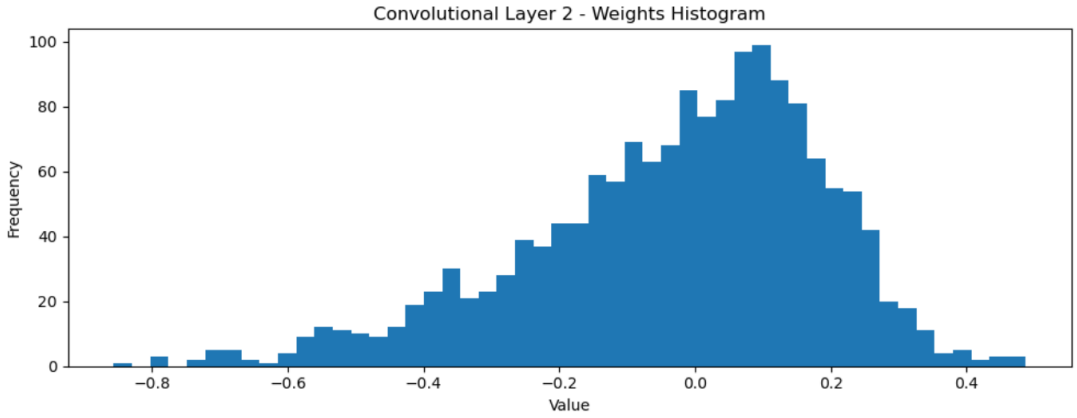
Conclusion:

Filter size 和 stride 對分類的正確性和 loss function 的值(預設為 cross entropy)並沒有顯著的影響。strides size 最主要的影響是，stride 越大則輸出的變數越少，Filter size 則影響進行卷積時擷取特徵的範圍大小，決定輸出大輪廓還是細節(影像存在的邊緣等等特徵)。

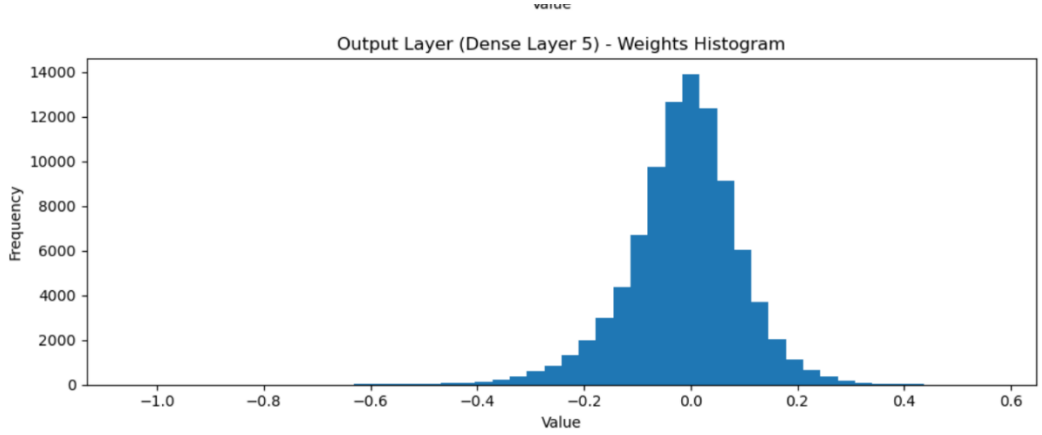
輸出尺寸 Output size = (Input size – Filter size)/Stride，故輸出由兩者共同決定，通常較大的 stride 會搭配較大的 filter。

以下 plot 以 filter size (5,5), strides(1,1)模型呈現

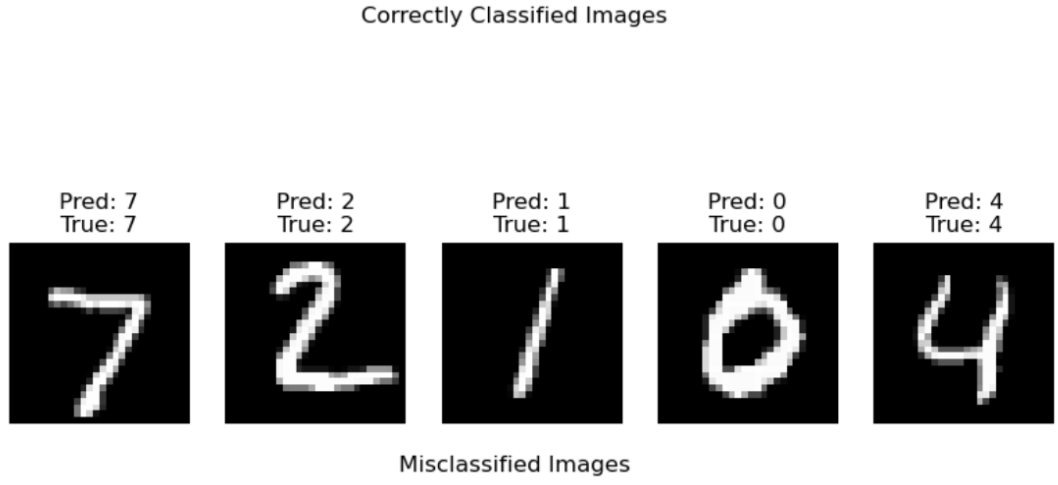
Weight histogram of conv. Layer 1



Weight histogram of dense layer(also output layer)



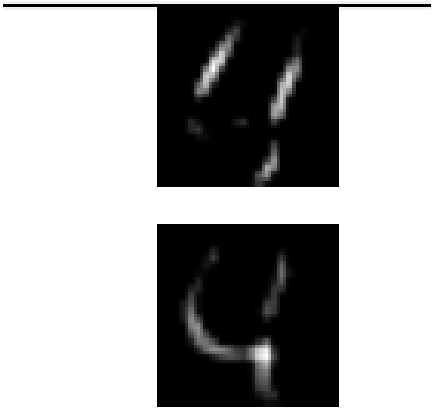
Some examples of correctly classified



Some examples of miss-classified

Observe the feature maps from different layers:

input layer

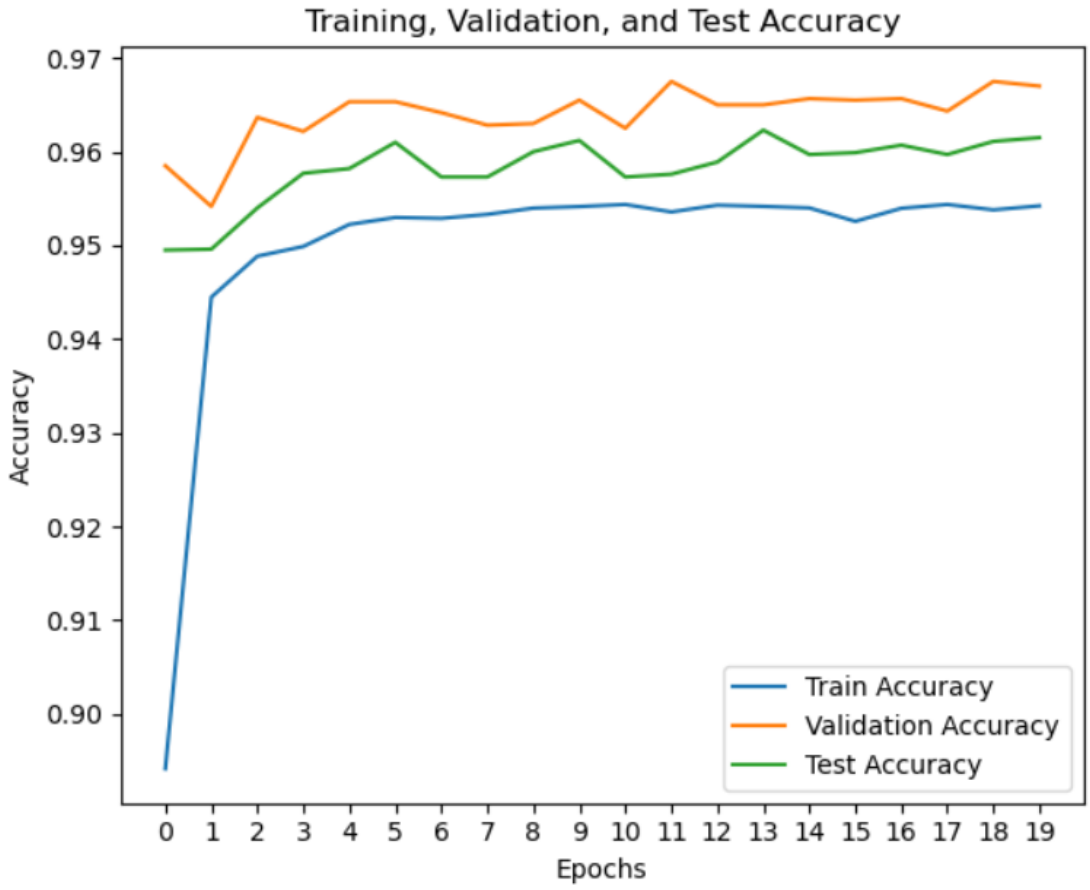


conv layer

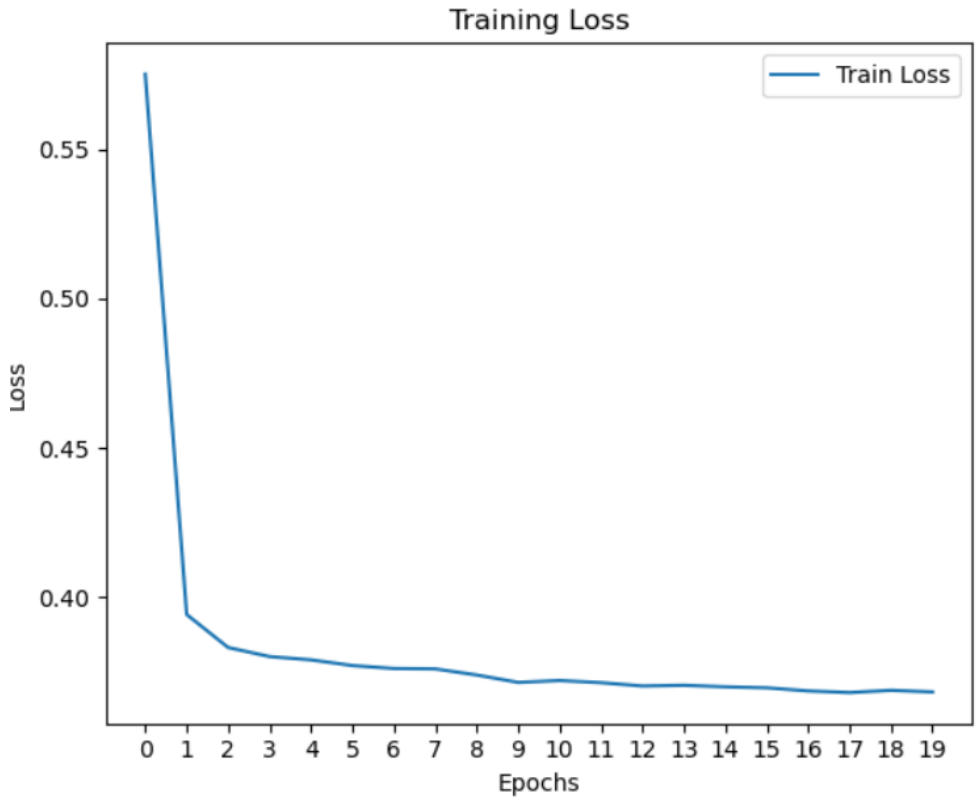
Max polling layer

Add L2 regularization:

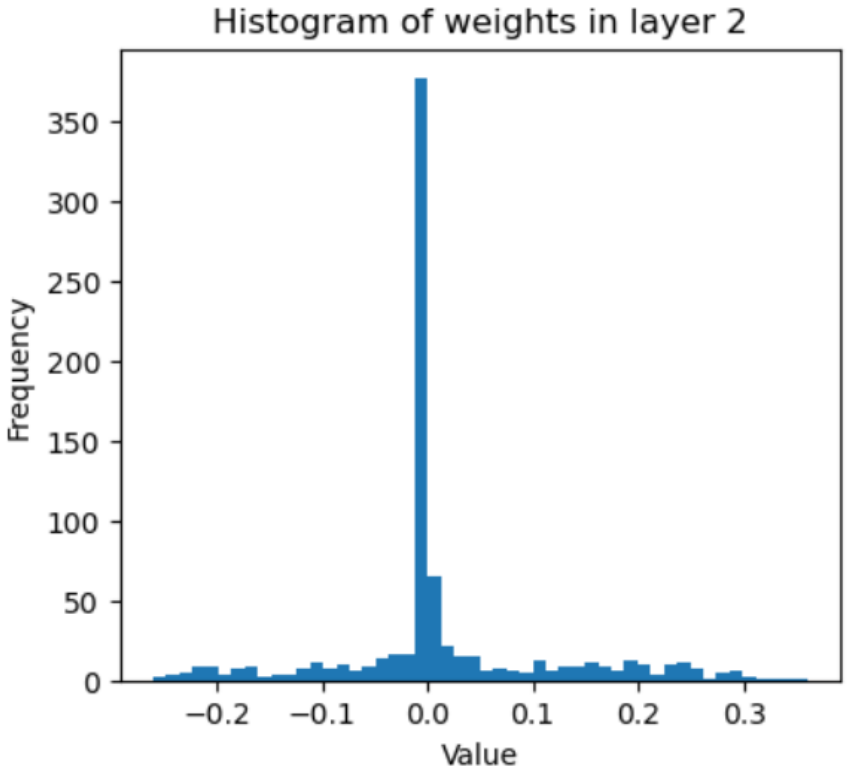
accuracy plot



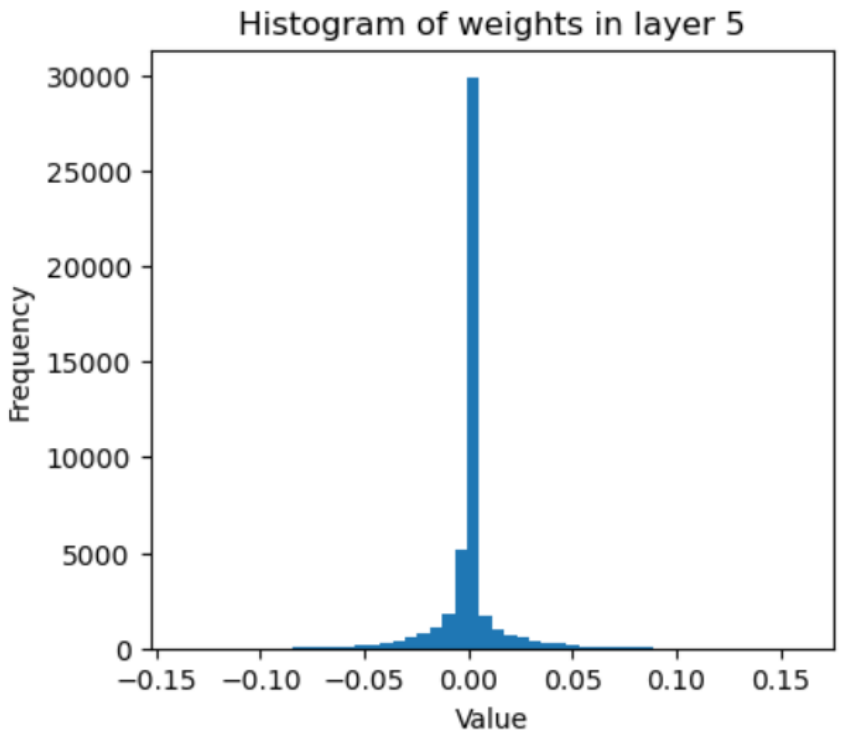
Loss plot



Weight histogram of conv1



Weight histogram of dense layer(also output layer)



Conclusion:

在模型中加入 L2 Regularization 以後，weights matrices 變得更為集中到 0 附近，是由於下面這個式子中

$$E = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} y_{nk} \ln t_{nk} + \alpha\|\omega\|_2^2$$

後半部 2nd norm 的額外得懲罰項（penalty term），用於抑制 weight W 變成較小的值，會讓模型變得較不易 overfitting 或是過度依賴某些特徵。在沒有加入 L2 regularization 前，test accuracy 和 valid accuracy 在 epoch = 3 以後都小於 train accuracy，但在加入 L2 regularization 以後就沒有這樣的現象(test and valid accuracy 還是高於 train accuracy)，顯示有適當的抑制 overfitting 現象。

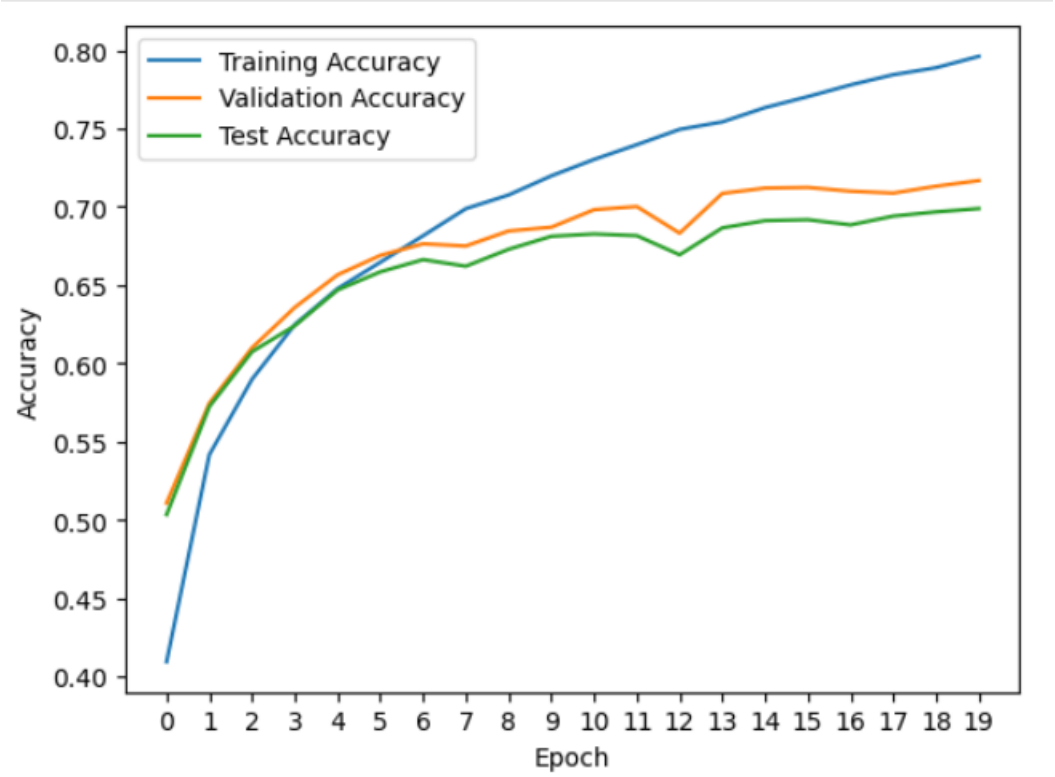
Part II. CIFAR

Convolutional Neural Network Structure:

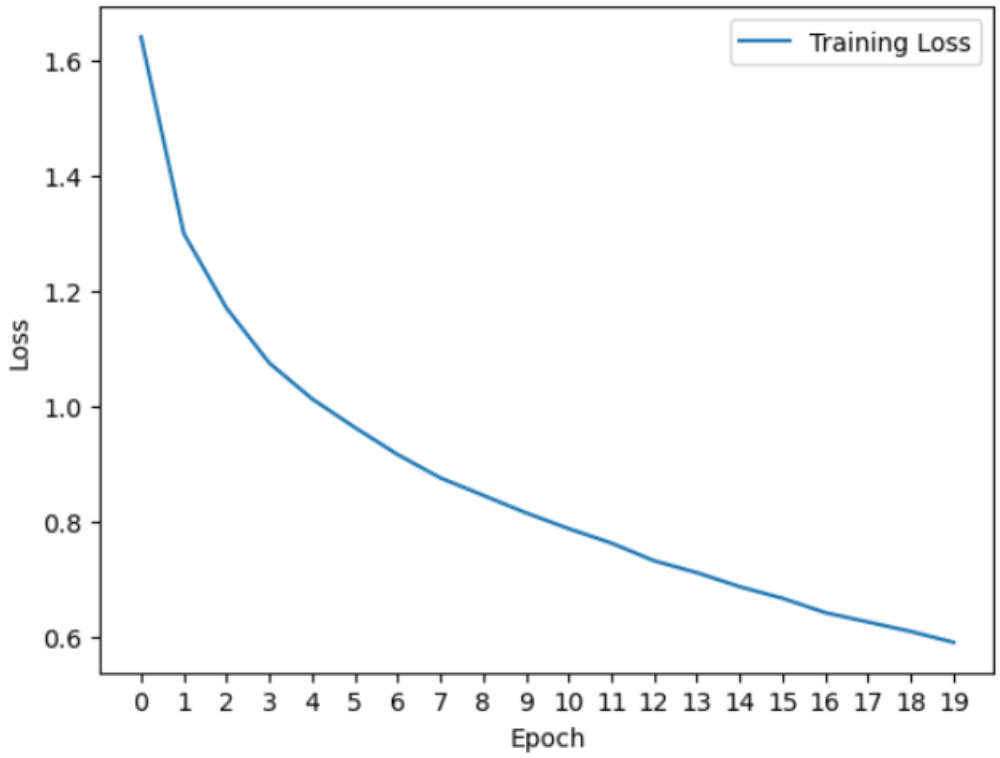
Filter size: (3,3), Stride:(1,1)

	Layer Name	Layer Type	Output Shape
0	conv2d_41	Conv2D	(None, 30, 30, 32)
1	max_pooling2d_31	MaxPooling2D	(None, 15, 15, 32)
2	conv2d_42	Conv2D	(None, 13, 13, 64)
3	max_pooling2d_32	MaxPooling2D	(None, 6, 6, 64)
4	flatten_13	Flatten	(None, 2304)
5	dense_26	Dense	(None, 64)
6	dense_27	Dense	(None, 10)

Accuracy Plot

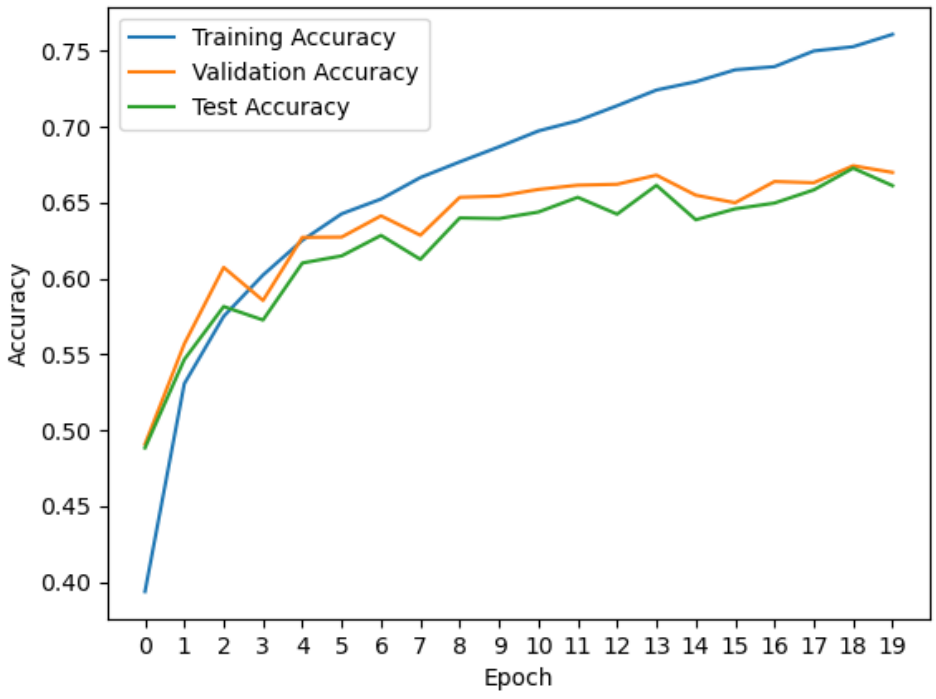


Loss

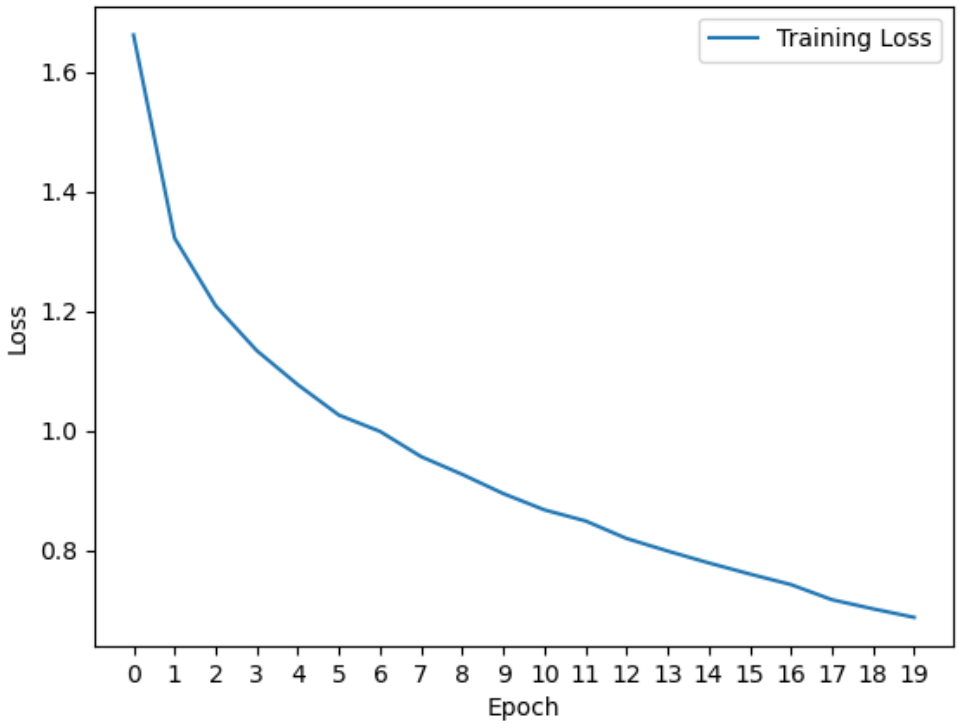


Filter size: (5,5), stride(1,1)

Accuracy



Loss function

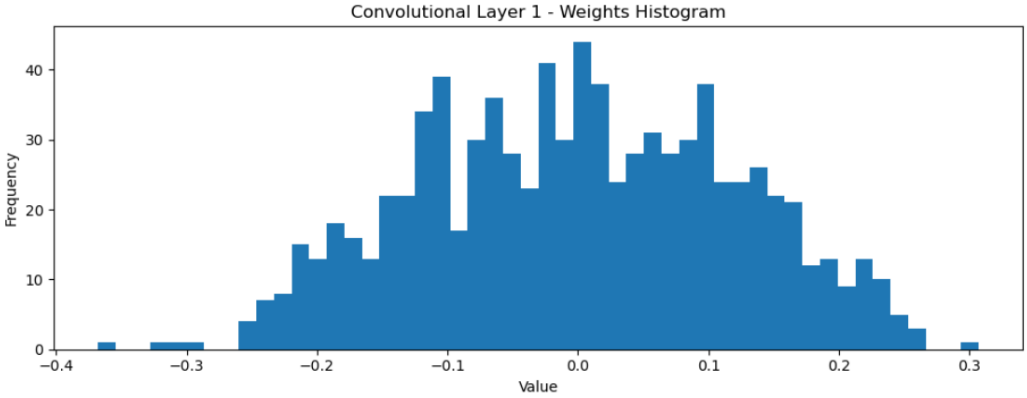


Conclusion:

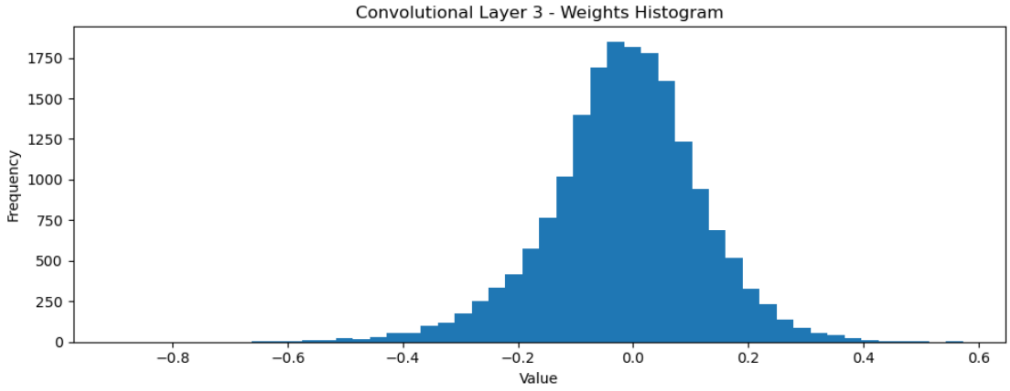
Filter size 和 stride 對分類的正確性和 loss function 的值(預設為 cross entropy)並沒有顯著的影響，當使用較大的 filter 時不論是 train accuracy, test accuracy 和 valid accuracy 都較低，不過以整體而言並沒有明顯影響訓練出的神經網路，和 part 1 MNIST 的結論相同，影響的應該是計算模型所需的時間而非分類結果。

以下 plot 以 filter size (3,3), strides(1,1)模型呈現

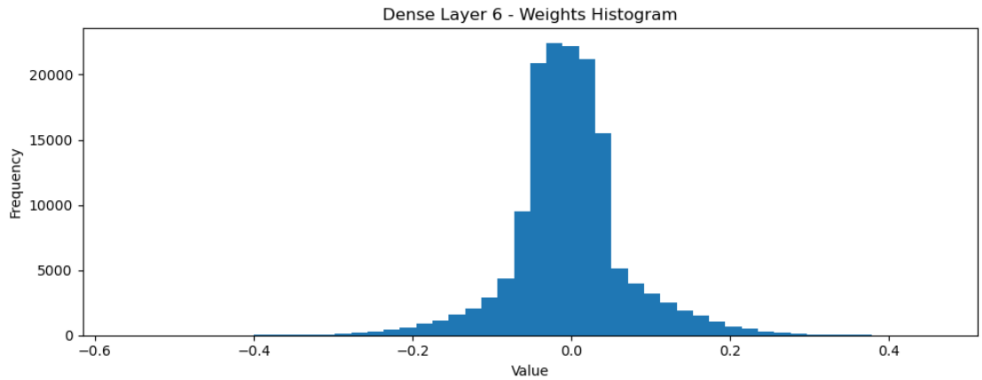
Weights Histogram of convolution layer 1



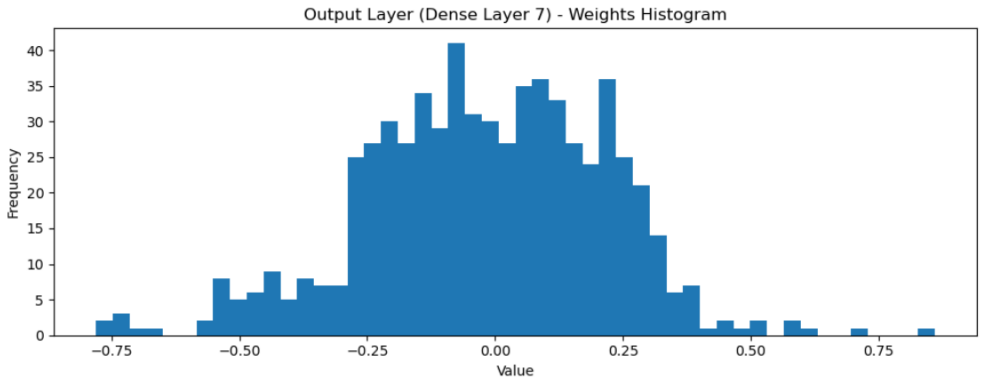
Weights Histogram of convolution layer 2



Weights Histogram of dense layer



Weights Histogram of dense layer 2(Output layer)



Some examples of correctly classified

Predicted: 8, Actual: 8

Predicted: 0, Actual: 0

Some examples of Incorrectly classified

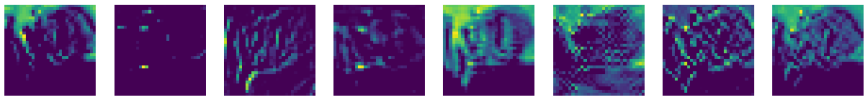
Predicted: 1, Actual: 8

Predicted: 4, Actual: 0

Observe the feature maps from different layers:

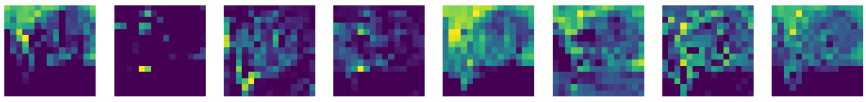
Convolution layer 1

Layer 1: conv2d



Max pooling layer 1

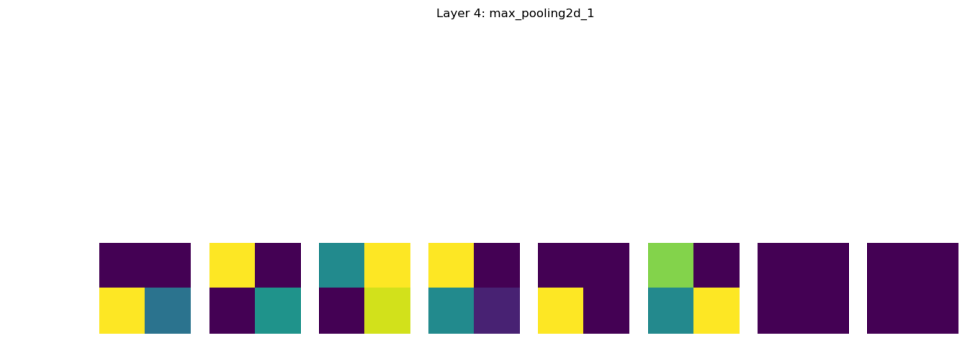
Layer 2: max_pooling2d



Convolution layer 2

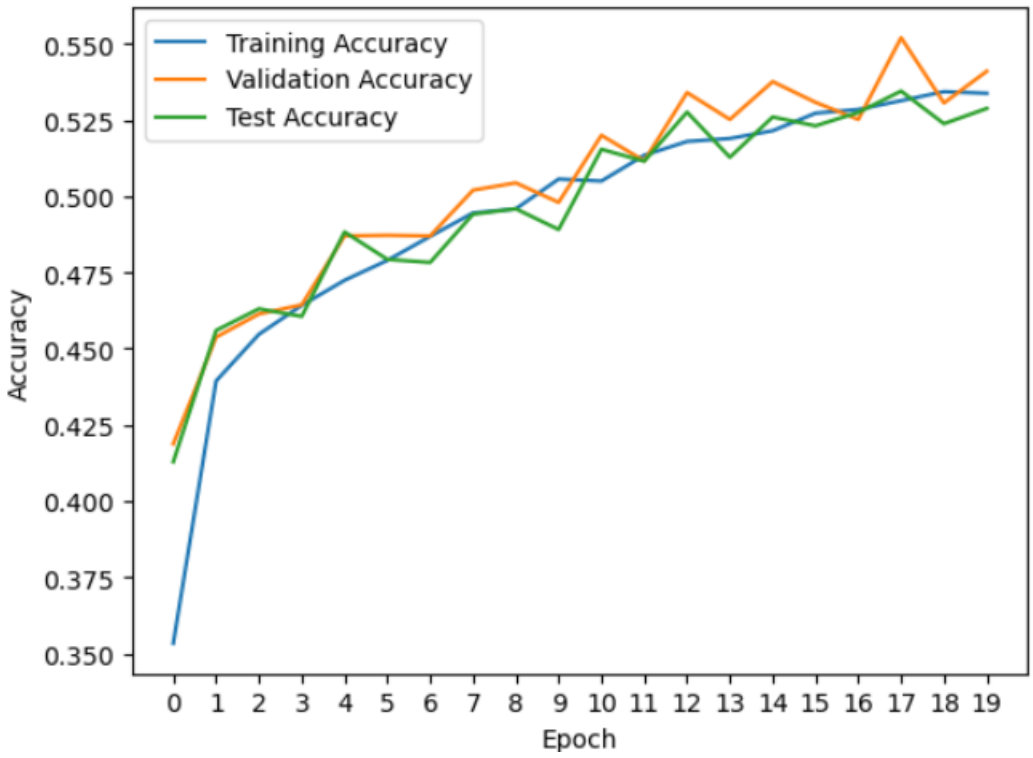
Layer 3: conv2d_1

Max pooling layer2

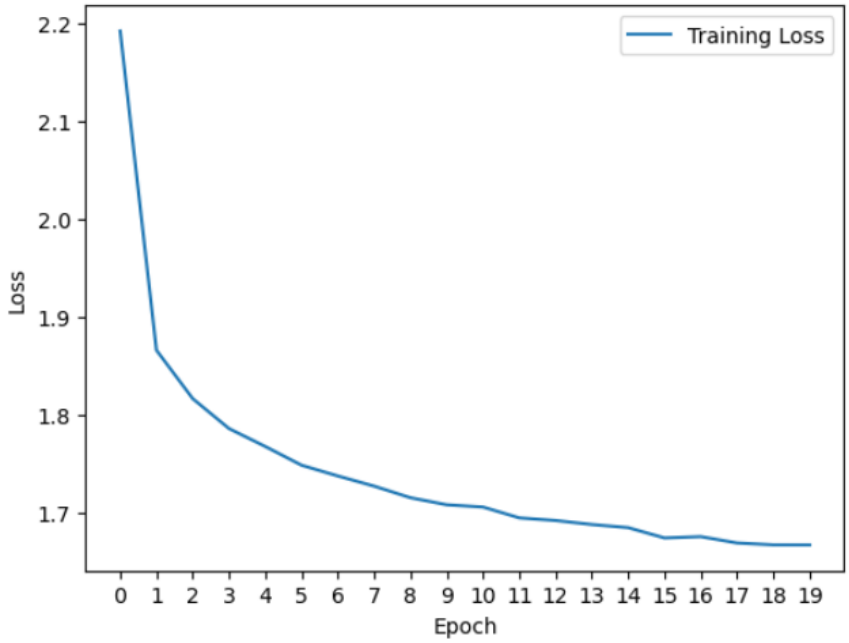


Add L2 regularization:

Accuracy plot



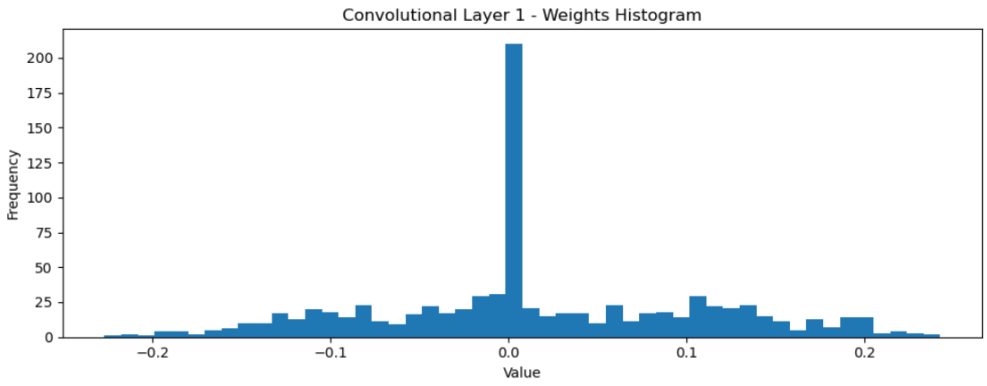
Loss plot



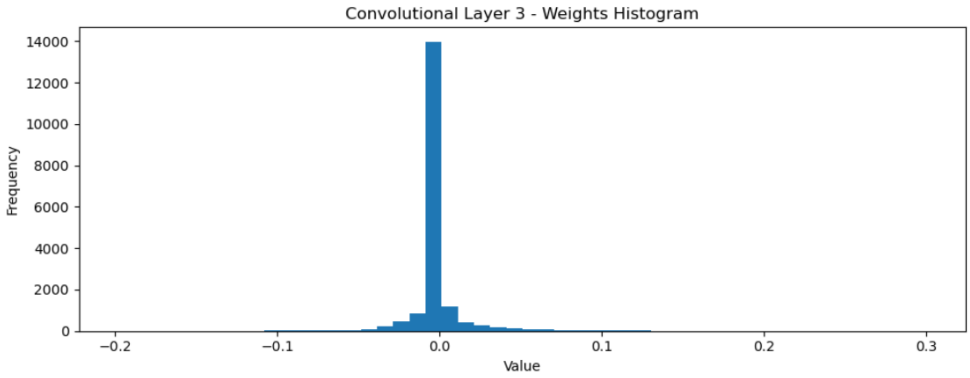
Conclusion:

在 CIFAR 10 所訓練出的模型上使用 L2 regularization 的效果並不好，推測原因如下：原本 L2 Regularization 的目的應該是防止訓練過程中 overfitting 發生，但原先未使用 L2 Regularization 的 model 正確率落在 60~70%之間，並沒有 Overfitting 的現象發生，然而在使用 L2 regularization 以後變成的 underfitting 的模型，分類 accuracy 明顯下降許多，接下來的幾張 weights histogram 也說明 L2 regularization 會產出複雜度較低的模型(大量 weights 接近 0)。

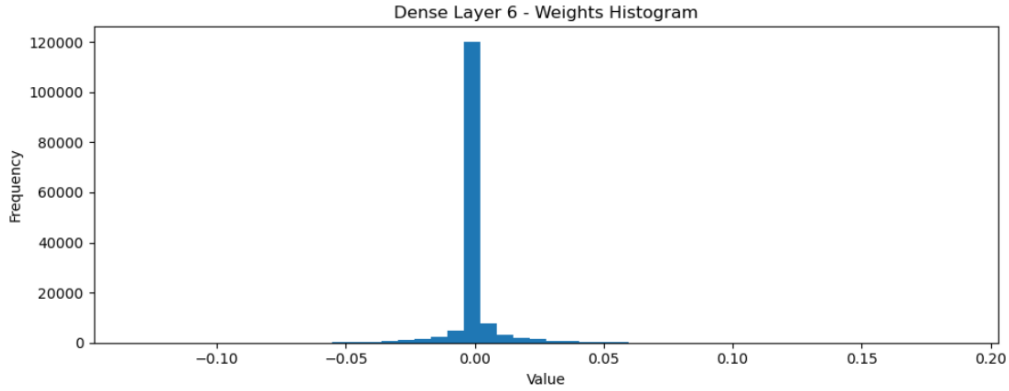
Weights histogram of convolution layer 1



Weights histogram of convolution layer 2



Weights histogram of dense layer 1



Weights histogram of convolution layer 2(output layer)