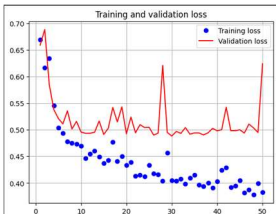
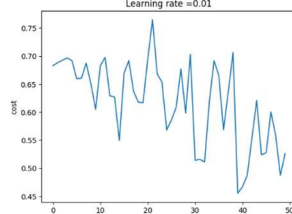
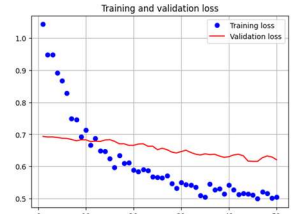


## Simple Experiment

	Linear Model	My CNN Model	TensorFlow CNN Model
Epochs	50	50	50
Training time	5.4s	97m9.5s	18.3s
Accuracy	Train:0.66 Valid:0.65	Train:0.74 Valid:0.71	Train:0.77 Valid:0.79
Number of parameters	32833	7221	64625
Training loss curve			

## Advanced part design:

在 advanced part 的 model 架構上，基本上是沿用 basic part 的部分，一樣是兩層 convolutional + pooling，但在 dense layer 上選擇了三層 relu 和最後一層 sigmoid。而在細節上，參數對應 basic 有做調整，在 convolutional layer 轉換到 dense layer 時，中間有做 Dropout 40% 的 units，避免 units 過多導致 overfitting，而在 dense layer 之間，都有做 batch normalization。最後在 compile model 時，optimizer 根據網路上最常用的是 adam 和 SGD 兩種，在經由多次嘗試後，最終仍選擇 adam 作為 model 的 optimizer，會這麼選擇可能有一部分原因是因為自己的英文名字就是 Adam 吧 XD。而 loss function 則因為最後一層只有一個 unit 且為 sigmoid，所以選擇的是 binary crossentropy。在資料處理上，我有先運用 data augmentation 將資料集擴充，但因為本身的 dataset 就沒有很大且圖片的 size 也偏小，為了保證 data 仍能辨識，能做的變化其實不多，而就在做完 data augmentation 後，初期的 training 反而 loss 升高，但到後期就逐漸升高，能 train 的 epochs 也變多，相比沒有做 augmentation 前，並不會那麼容易就 overfitting，整個 model 的泛性有所提高。

## Summary:

此次 CNN 的作業相比前幾次打 code 的量又變得更少了，基本上都是建立於上一次作業的 neural network，再在前面增加 convolutional network 而已，而且在實作上，因為沒有要讓我們實作 backpropagation，僅有實作 forward 和 initial 等較為簡單的事情，並且 template 中給出的資訊再搭配上課講義，在打 code 上面的進度是比較順利的，只是因為圖片維度在經過 filters 之後會有所改變，這部分的維度對應上卻也花了不少時間，但若真的要講到最花時間的事情，應該是 CNN 是如何運作的原理吧，因為其想法有點過於抽象，要說服自己那就是一堆數字轉

換來轉換去，最後得出一個結果，這個過程是不太容易的。

而後 advanced 的部分，因為可以使用 TensorFlow，在架構 model 的部分上更是快速，在 google 如何使用 TensorFlow 去建立 CNN 後，按照網上的教學大概幾十分鐘內就寫好了，但為了要使用 GPU 加速，架設環境上卻是煞費苦工，前前後後大概花了好幾個小時才將環境架好，各種 package 的衝突和版本不合，安裝了又卸載，反反覆復真的一度想要放棄，後來想到多花點時間，省下的卻是 training 那漫長的等待過程，就咬著牙繼續架設，所幸最後還是有成功，看到 training 的時間被壓縮了幾十倍，頓時覺得自己也是一個 ML 的工程師了。

此次的作業不僅讓我接觸到時下最熱門的 CNN model，也就是電腦視覺的部分，也讓我開始使用 python community 的 package 後，感覺好像已經成為了這個圈子的一份子，覺得倍感成就，接下來所剩差不多一個月的時間要與組員們一起努力奮鬥 final project，也期待可以在這門課上完美的畫下一個句點。