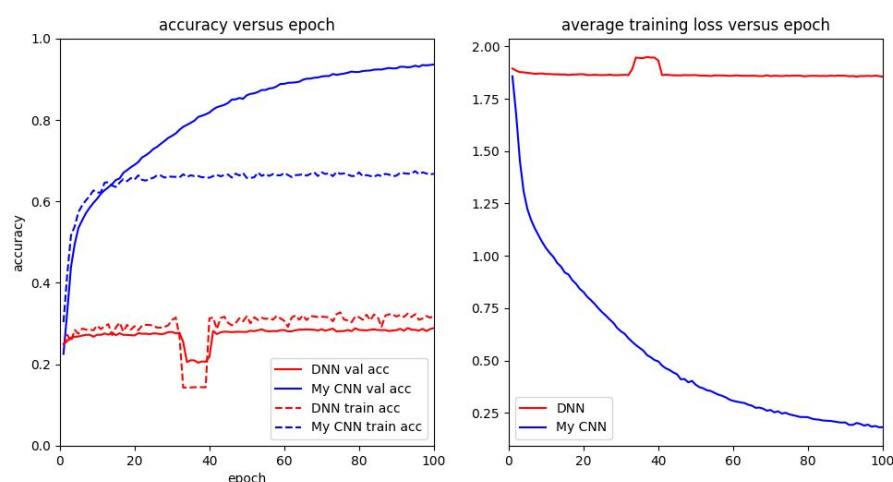


1. (2%) 請說明你實作的 CNN model，其模型架構、訓練參數和準確率為何？並請用與上述 CNN 接近的參數量，實做簡單的 DNN model，同時也說明其模型架構、訓練參數和準確率為何？並說明你觀察到了什麼？(Collaborators: No)

答：我用的模型架構參考VGG16，共有六層，前五層每層由是兩個kernel size=3*3的convolution layer與一個maxPooling組成，最後接了兩個fully connected network。這個模型在kaggle public與private測資的成績分別為0.6896/0.68877。我的CNN參數量為546529，因此我實做一個參數量為547707的 $48^2 \times 237$ 、 237×7 的簡單兩層DNN，此DNN幾乎train不起來，在100epoch後training accuracy僅0.28，在kaggle pubic/private的成績為0.33324/ 0.33491。因此可以發現CNN的filter設計能有效的捕捉feature。

2. (1%) 承上題，請分別畫出這兩個model的訓練過程 (i.e., loss/accuracy v.s. epoch) (Collaborators: No)

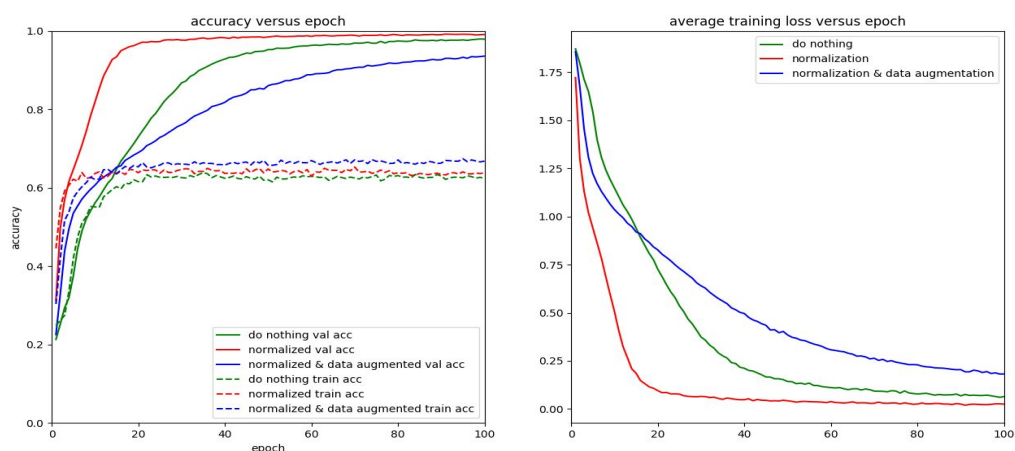
答：如下圖，可以發現DNN的loss居高不下，而準確度也一直沒有上升。



3. (1%) 請嘗試 data normalization, data augmentation, 說明實作方法並且說明實行前後對準確率有什麼樣的影響？

(Collaborators: No)

答：我實作data normalization的方法為先對原圖像除以256，並且在NN模型中，每通過一個Activation function後就引入一個nn.BatchNorm2d或nn.BatchNorm1d；實作data augmentation的方法則是透過：在自訂的dataset class的__getitem__()對要被取用的圖片以torchvision.transforms做一系列的轉換後才return來做。圖片在被取用前，會先被ColorJitter()隨機改變明暗、對比與飽和度，RandomRotation()隨機轉一點角度、RandomResizedCrop()隨機做剪裁再放大回原本大小，才return給data_loader訓練模型。



由實驗結果圖發現，可以在使用data normalization前模型在training accuracy與validation accuracy的上升速度都是最慢的，並且在validation accuracy收斂後，得到的準確率最低，約為0.62；在使用data normalization後，training accuracy上升速度變快，但是在validation accuracy收斂後與沒做normalization相比，僅有小幅領先，停滯在0.64左右。再加上使用了data augmentation後可以發現training accuracy上升的較慢，但是validation accuracy上升至約0.68左右，這代表使用data augmentation能讓模型的overfitting程度降低，並增加validation的精準度。

4. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析](Collaborators: No)



答：

我將testing set切1/5作為validation set 作圖。模型易將0(生氣)辨識為2(恐懼)或4(難過)，2(恐懼)辨為4(難過)，4(難過)辨為0(生氣)，2(恐懼)或6(中立)，6(中立)辨為4(難過)。彼此容易搞混的class為生氣、恐懼、難過，難過又容易與中立搞混，這些表情都有類似的特徵(嘴往下彎)，就算我辨識也不一定正確。最容易分辨的是高興、驚訝，由於兩者嘴形較為明顯，因此合理。