National Tsing Hua University
Fall 2023 11210IPT 553000
Deep Learning in Biomedical Optical Imaging
Homework 2

林伯諭

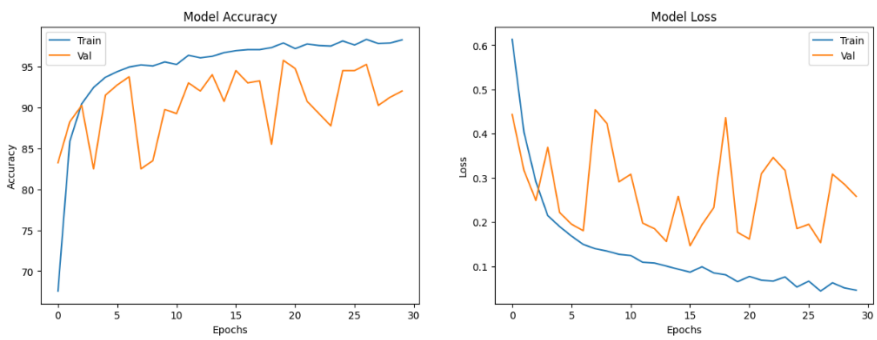
Student ID: 111066538

1. Coding

1.1 Task A: Transitioning to Cross-Entropy Loss

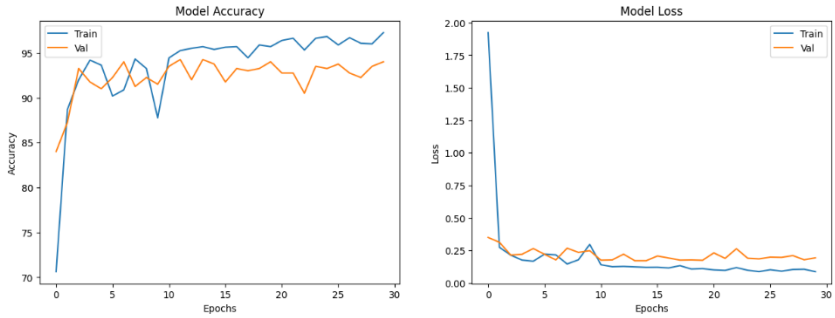
要將 BCE 改成 CE 首先要先替換 LOSS FUNCTION，他可以從 PyTorch 中找到，需要使用 CrossEntropyLoss 去將原本的 BCEWithLogiteLoss 替換掉，但是只將 LOSS FUNCTION 改掉程式的輸出會只有 0，那是因為 BCE 與 CE 有個最大的差距就是 BCE 是二元分類所以必須將最後一層的 NODE 改多，並且 outputs 是一個[16，16]的方陣所以必須將其降階到[16]的 array 才能與 labels 做比較

1.2 Task B: Creating a Evaluation Code



圖一

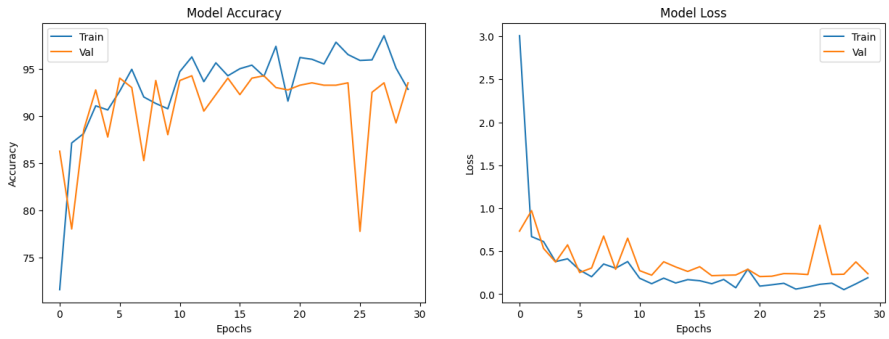
圖一是用原本 lab3 的程式跑出來的結果，可以發現他的 Train Loss 有再下降但是 Val Loss 卻沒甚麼變化，還有 Train Accuracy 以及 Val Accuracy 之間的差距蠻大的，那可能是因為這個架構 over fitting 了，所以我將 neural network 的層數降為一層，並重新訓練一次並記錄在圖二，可以發現 Train Accuracy 與 Val Accuracy 之間的差距縮小了，並也可以看到 Val Loss 也有慢慢下降的跡象



圖二

2. Report

2.1 Task A: Performance between BCE loss and BC loss



圖三

圖三是訓練單層 CE 所得到的圖，因為用來比較的 BCE 以及 CE 的 neural network 皆只有一層，並且使用相同的 learning architecture 還有 hyperparameters， 並將前三次訓練以外的所有的 Train Loss、Train Accuracy、Val Loss 以及 Val Accuracy 做平均值可以得到表一:

表一

	Train Loss	Train Accuracy	Val Loss	Val Accuracy
BCE	0.1845	98.68%	0.2723	93.43%
CE	0.1339	94.87%	0.2042	92.81%

並且在各經過 3 次訓練後可以得到表二:

表二

	Train Loss	Train Accuracy	Val Loss	Val Accuracy
BCE	0.0365	98.68%	0.2723	93.43%
CE	0.0590	97.75%	0.2333	93.49%

可以看到 BCE 的 Train Loss 在經過四輪訓練後是呈現下降的趨勢，而 Val Loss 並沒有太大的變化，而且 Train Accuracy 以及 Val Accuracy 也沒太大的變化，這意思就是他只需要第一輪的訓練就能有不錯的成果。相較之下 CE 的 Val Loss 有些許的上升，那可能是因為有 over fitting 的情況產生，但是從結果來看 Train Accuracy 從 94.87%上升到 97.75%約上升了 3%，又 Val Accuracy 從 92.81%上升到 93.49%也上升了些許，所以我認為該模型有些許的改善，在此可以做個小結論就是在相同架構下，CE 的訓練次數需比 BCE 多一點。

2.2 Task B: Performance between Different Hyperparameters

第一個我要改的 hyperparameter 是 batch size 原本的大小為 32，而我會使用 16 以及 64 來測試，並與上面一樣將前三次訓練以外的所有的 Train Loss、Train Accuracy、Val Loss 以及 Val Accuracy 做平均值，這邊我使用的架構是單層的 BCE，其比較表格為表三:

表三

	Train Loss	Train Accuracy	Val Loss	Val Accuracy
batch size=32	0.1845	98.68%	0.2723	93.43%
batch size=16	0.1342	94.84%	0.2325	92.26%
batch size=64	0.1275	95.07%	0.2334	92.31%

並且在各經過 3 次訓練後可以得到表四:

表四

	Train Loss	Train Accuracy	Val Loss	Val Accuracy
batch size=32	0.0365	98.68%	0.2723	93.43%
batch size=16	0.014	99.64%	0.3341	93.57%
batch size=64	0.0583	98.25%	0.2345	93.36%

可以從表三看到 batch size 的大小並不會改善第一次的學習狀態，但也可以從表四看到多次訓練後較小的 batch size 可以有較高的 Train Accuracy，而 Val Accuracy 也有些許的上升。

第二個我要改的 hyperparameter 是 Activation Functions，通過單層 BCE 來測試 ReLU、Leaky ReLU，並與上面一樣將前三次訓練以外的所有的 Train Loss、Train Accuracy、Val Loss 以及 Val Accuracy 做平均值，並做出表五:

表五

	Train Loss	Train Accuracy	Val Loss	Val Accuracy
ReLU	0.1845	98.68%	0.2723	93.43%
Leaky ReLU	0.1329	94.97%	0.2636	91.60%
Tanhchrink	0.2065	94.85%	0.3385	92.29%

並且在各經過 3 次訓練後可以得到表六:

表六

	Train Loss	Train Accuracy	Val Loss	Val Accuracy
ReLU	0.0365	98.68%	0.2723	93.43%
Leaky ReLU	0.0953	96.39%	0.2421	92.93%
Tanhchrink	0.0738	97.49%	0.3046	92.93%

從表五、六可以看到 ReLU 的正確率是最高的，但是由於他的 Val Loss 沒有太大的變化然而其他兩個還有下降的趨勢，因此在訓練第 4 次後可以得到表七數據:

表七

	Train Loss	Train Accuracy	Val Loss	Val Accuracy
Tanhchrink	0.03488	99.08%	0.4317	93.5%

可以發現它的正確率又比之前高了。