

National Tsing Hua University

Fall 2023 11210IPT 553000

Deep Learning in Biomedical Optical Imaging

Homework 4

林伯諭

Student ID: 111066538

1. Task A: Model Selection

我使用的兩個架構分別是 resnet101 以及 Resnext50_32x4d。可以從表一看到 resnet101 與 Resnext50_32x4d 都架構非常複雜。resnet101 使用了四個 Layer 並且四個 Layer 的 Bottleneck Layer 層數並不是相同的，從第一層到第四層分別為 3、3、23、3，而 Bottleneck Layer 的好處是可以降低運算量，並且可以改變維度，而其每個 Bottleneck Layer 都是由三個 Conv2d 所組成，而三個 Conv2d 中的第二個其 kernel size 一定比第一個以及第三個大那是做 Bottleneck Layer 的必要結構之一。而 resnext50_32x4d 的組成與 resnet101 相似，他們的 Layer 數以及 Conv/ Bottleneck 數是相同的，但是他的 Bottleneck Layer 比 resnet101 少了一點，那可能就是導致 resnext50_32x4d 的運行時間比 resnet101 少了一分鐘。

從表二的第二欄可以看到 resnet101 以及 resnext50_32x4d 的 Best Train acc 皆達到了 100%，並從 Best Val Loss 那一欄以及最佳 Val Loss 位置以及數值那一欄可以看到，resnet101 有 Overfitting 的狀況，雖然 Resnext50_32x4d 也有此症狀，但是比起 resnet101 好了一點，那可能是因為 resnet101 的 Bottleneck Layer 比 resnext50_32x4d 複雜了一點，而越複雜的架構越容易使訓練 Overfitting，因此若要改善這兩個架構。我認為要先讓他不那麼貼合訓練的樣本。

表一

	Layer 數	Bottleneck/Layer	Conv/ Bottleneck	花費時間
resnet101	4	3->3->23->3	3	16 分鐘
resnext50_32x4d	4	3->4->6->3	3	15 分鐘

表二

	Final Train Loss	Best Train acc	Best Val acc	Final Val Loss	最佳 Val Loss 位置以及數值
resnet101	0.0465	100%	98.25%	0.1029	Epoch 16/30(0.0358)
resnext50_32x4d	0.0221	100%	99.75%	0.1724	Epoch 20/30(0.0103)

2. Task B: Fine-tuning the ConvNet

可以從表三以及表二看到，在使用 Transfer Learning 之後，它的性能有這顯著的提升從比較中可知，Best Val Loss 雖然沒有太大的改變，但是他的 Overfitting 已經降低了一點，尤其是 resnext50_32x4d 在 30 次訓練後他的 Final Val acc 還在下降，並且 Test accuracy 上升了不少 resnet101 以及 resnext50_32x4 分別上升了 6.5 趴以及 3.25 趴。而且這不只是單一個案，因為使用的兩個函數 resnet101 以及 resnext50_32x4d 都有改善。

表三

	Final Train Loss	Best Train acc	Best Val acc	Final Val Loss	最佳 Val Loss 位置以及數值	Test accuracy
resnet101	0.0002	100%	98.5%	0.0501	Epoch 23/30(0.0441)	79%
resnext50_32x4d	0.0007	100%	99.75%	0.0117	Epoch 30/30(0.0117)	76.75%

並且我改變了 resnext50_32x4d 的 weights，從原本的 IMAGENET1K_V1 改成 IMAGENET1K_V2，同時我也將 resnet101 的 weights 改為 IMAGENET1K_V2，並將兩數據記錄在表四：

表四

	Final Train Loss	Best Train acc	Best Val acc	Final Val Loss	最佳 Val Loss 位置以及數值	Test accuracy
resnet101	0.0002	100%	99%	0.0623	Epoch 15/30(0.0378)	77.5%
resnext50_32x4d	0.0002	100%	99.75%	0.0094	Epoch 29/30(0.0059)	80.75%

可以從表四觀察到更換 weights 的結果不一定相同，可以看到 resnet101 他的 Best Val acc 上升了一點，然後最佳 Val Loss 位置以及數值提前了一點，還有 Test accuracy 卻下降了百分之 1.5，這就代表 resnet101 使用 IMAGENET1K_V2 後會加重 Overfitting 的狀況。但是 resnext50_32x4d 將 weights 更改為 IMAGENET1K_V2 後卻把效能變好了，他的 Test accuracy 從 76.75% 上升到 80.75%，整整上升了 4 個百分比，因此可以判斷要改善不同的架構皆須使用不同的 weights。

3. Task C: ConvNet as Fixed Feature Extractor

表五

	Final Train Loss	Best Train acc	Best Val acc	Final Val Loss	最佳 Val Loss 位置以及數值	Test accuracy
resnet101	0.1028	96.5%	95.75%	0.1181	Epoch 30/30(0.1158)	82%
resnext50_32x4d	0.0903	96.81%	94.75%	0.1068	Epoch 17/30(0.1068)	86.25%

從表五可以看到相較於使用第一種方法，將訓練的地方凍結住效能更好，並且可以發現這個方法將大大縮短運行時間，resnext50_32x4d 從 15 分鐘縮短到了五分鐘又十秒，這意思是他只要原本程式的三分之一，而 resnet101 也節省了很多時間，從 16 分鐘減少到六分二十四秒，約為二點五分之一。並且可以看到 resnext50_32x4d 有些許的 Overfitting，而 resnet101 卻沒有。

4. Task D: Comparison and Analysis

表六

	Final Train Loss	Best Train acc	Best Val acc	Final Val Loss	最佳 Val Loss 位置以及數值	Test accuracy
resnet101(B)	0.0002	100%	98.5%	0.0501	Epoch 23/30(0.0441)	79%
resnet101(C)	0.1028	96.5%	95.75%	0.1181	Epoch 30/30(0.1158)	82%

表七

	Final Train Loss	Best Train acc	Best Val acc	Final Val Loss	最佳 Val Loss 位置以及數值	Test accuracy
resnext50_32x4d(B)	0.0007	100%	99.75%	0.0117	Epoch 30/30(0.0117)	76.75%
resnext50_32x4d(C)	0.0903	96.81%	94.75%	0.1068	Epoch 17/30(0.1068)	86.25%

表六以及表七分別為 resnet101 在 B 架構以及 C 架構的詳細資料，表六則是 resnext50_32x4d 在 B 架構以及 C 架構的詳細資料，兩個的 B 架構都是使用 IMAGENET1K_V1。首先從表六可以看到 B 的 Best Train acc 已經達到了 100%而 C 架構卻只有 96.5%，並且從最佳 Val Loss 位置以及數值那一欄可以看到，B 架構的 Val Loss 從 Epoch 23/30 開始就沒有再下降了，結合這兩個數據可以判斷 B 已經 Overfitting 了。現在看看表七，Best Train acc 與表六一樣都是 B 架構比較高，但是這次可以看到最佳 Val Loss 位置以及數值與表六相反，雖然有 Overfitting 的情況產生，但我認為 C 類型的架構不需要管這方面的事情，因為當設計者 freeze 住架構時 backward()不會運作，這就導致沒有反向傳播去修正梯度。並且我們可以從表五以及表七同時看到 Test accuracy 在 C 架構都比 B 架構好，而且這個好壞程度是不管有沒有 Overfitting，因此我想那可能是因為 input 的資料與測試的資料有很大的不相關性，導致無法正確的判斷對錯。並且可以從表六的 Test accuracy 欄以及 Best Val acc 欄看到，當 Best Val acc 上升 Test accuracy 就會下降，這在 resnext50_32x4d 中也能看到，我想那可能是因為在 Transfer Learning 中較好的 Val acc 代表他很貼合訓練的模組，因此導致測試集無法正確的判別。並且可以看到 C 架構的 Test accuracy 皆比 B 架構高，那可能是因為要判斷醫療照片 C 架構比 B 架構較適合。

5. Task E: Test Dataset Analysis

表八

	Final Train Loss	Best Train acc	Best Val acc	Final Val Loss	最佳 Val Loss 位置以及數值	Test accuracy
rsnet18(out features=10)	0.0000	100%	99.75%	0.0185	Epoch 26/30(0.0152)	77.5%
rsnet18(out features=2)	0.0000	100%	99.5%	0.0107	Epoch 19/30(0.0098)	78.5%
rsnet18(out features=6)	0.0001	100%	99.25%	0.0260	Epoch 9/30(0.0098)	76.25%

表九

	Final Train Loss	Best Train acc	Best Val acc	Final Val Loss	最佳 Val Loss 位置以及數值	Test accuracy	time
rsnet18(x_train:1 3 1 1 ;x_val:1 3 1 1)	0.0000	100%	99.5%	0.0107	Epoch 19/30(0.0098)	78.5%	3:10
rsnet18(x_train:1 3 3 1 ;x_val:1 3 3 1)	0.0001	100%	99.5%	0.0274	Epoch 25/30(0.0196)	71.25%	10:08
rsnet18(x_train:1 3 3 1 ;x_val:1 3 1 1)	0.0002	100%	99.5%	0.0455	Epoch 10/30(0.0318)	82.25%	9:17

在這邊我修改了兩中參數來看 rsnet18 的效能變化，首先表八是修改 out features 的值，將其定為 10 以及 6，並與原本的 2 做比較，可以看到 out features=10 的 Best Val acc 是最高的，但是他沒有改善 Test accuracy，並且 out features=6 的 Test accuracy 以及 Best Val acc 都是最低的，這就代表要改善效能不能從 out features 來下手。因此我打算改 x_train 以及 x_val，因此我使改了兩個數值，第一是將其改為 x_train:1 3 3 1&x_val:1 3 3 1，第二個則是 x_train:1 3 3 1&x_val:1 3 1 1，這兩個數據都可以從表九看到，可以發現同時將 x_train 與 x_val 改為相同的值，並不會改善效能，並且有可能使效能下降，因為當 x_train 改為 1 3 3 1 同時將 x_val 改為 1 3 3 1 他的 Test accuracy 會大幅的下降，從 78.5% 下降到 71.25%，並且使時間大幅上升從三分鐘上升到了十分鐘，增加了大概三倍。但是如果只改變 x_train，Test accuracy 就會有非常好的改變，當我將 x_train 改為 1 3 3 1 但 x_val 維持不變，會發現 Test accuracy 從 78.5% 上升到了 82.25%，其有非常顯著的改善，但是時間卻高達九分鐘。因此可以下一個結論就是改變 x_train 會使訓練時間上升，並且當 x_train 與 x_val 有不同的數值會有顯著的效果。

但是可以發現就算參數修改得再好，Test accuracy 也不會有九十趴以上的正確率。那可能是因為數據不匹配所造成的，因為拿一般的影像去判斷醫療影像，可想而知就是不好判斷，因此 Test accuracy 的正確率才不會變得太高。