

Homework 3 Report

學號：r06631035 系級：生機所碩二 姓名：王凱陞

1. 選擇使用 `pytorch` 進行這次的做參數調整作業，以下根據 `train.py` 的資料格式進行報告。

get random seed:

做了些微的測試，發現在不調動其他參數下開啟 `random seed` 的影響並不是很大，所以我這部分我先採取 `seed` 設定為 1 進行其他參數調整。

get model spec:

`pretrain` 部分我選擇直接啟用，也確實在這部分 `pretrain` 就有相當好的結果，而 `model name` 的部分，為求方便我直接在程式中加入一些常做調整的參數修改區，`model name` 即是直接將兩個 `model name` 存為陣列，方便切換。

get optimizer:

此部分我先讓兩個 `model` 分別針對 `SGD` 和 `adam` 這兩種我較常使用的 `optimizer`，發現在 `mobilenetv2` 中 `Adam` 表現較好，於 `resnet` 中則是 `SGD` 叫好。

get eval spec:

這部份最一開始，我是根據 `before_epoch` 做了哪些 `transform` 的 `data augmentation` 便作相同的 `transform`，因此導致 `validation` 結果都有點過高於 `kaggle` 的分數，所以最後索性改為完全不做其他處理。

before epoch:

這部份所選用的 `transform`，我先設想一個判斷圖片場景會有那些變化會增加辨識性，因此我選擇了 `RandomHorizontalFlip`、`RandomRotation`，稍微調一下參數就已經足以通過 `simple baseline`，此後我也繼續進行測試，加入了 `ColorJitter`，進行圖片的明亮度、對比度、飽和度變化，但這部份因為沒有足夠的影像處理基礎，所以值得調設上都調的較小一些，在 `batch_size` 的部分我將其由 32 起步，每 10 個 `epoch` 進行一次放大，並隨著每 10 個 `epoch` 的放大擴大增加的幅度，並在 64 時 `bound` 住，但經過測試之後發現以上作法並無法通用到兩個 `model` 上，以下為兩個 `Model` 分別所做的 `transform`。

Mobilenetv2 :

`Resize(224)`、`RandomHorizontalFlip`、`RandomRotation(20)`、`RandomAffine(20)`、`ColorJitter(brightness=0.1, contrast=0.25, saturation=0.25, hue=0.1)`、`ToTensor`。

Resnet50:

`Resize(224)`、`RandomHorizontalFlip`、`RandomRotation(18)`、`RandomAffine(18, translate=(0.2, 0.2))`、`ToTensor`

上述參數皆為由小到大逐漸測試上去，且也發現 `batch_size` 放大和 `learning rate` 下降的方式，對於 `resnet` 效果也不太好，所以只在 `mobilenet` 使用。

before batch:

這部分加入了 `learning rate decay`，透過每 15 個 `epoch` 增加一個 `cof` 參數，以 10 的 `cof` 次方進行 `learning rate decay`，在 `batch normalization` 的決定上一開始用了 0.3，但是在最後幾天測試下，發現調到 0.5，`validation` 分數有稍微提升，便加到 0.5，`dropout` 則是直接依據 `hw3` 在實作中，便發現 0.5 是一個滿有效的 `dropout` 比例直接用上。

save model as:

這部分花了些時間了解，因為過去沒使用過 `Pytorch`，不太了解其存 `Model` 的方式，所以我用了較為沒效率的方式，我將每次的 `epoch` 都做存檔，並計算哪個 `validation history` 最好，再人工的方式做 `inference`，不過之後為了方便辨識，也加入了時間和 `epoch` 次數到檔名內，方便之後把最好的幾個 `model` 作排列組合衝分數。(但這個決定，非常加深了重現的難度...)

2. 檢討與結論:

除了上述存檔方式造成重現非常困難以外，在 `random seed` 部分沒有設定也是個很大的問題，重現上也造成了部分的困難，且對於兩個模型也認識不足，導致調參數時，無法根據兩個模型的特性進行完善的調整，而是比較像時間換取分數的方式，稍微可惜，但在此作業中看到實際工作上是如何進行一層一層程式的規劃和資料結構的排序，在調參數時確實比自己胡亂打出的程式容易許多，此部分相當受獲，也很幸運的雖然 `public score` 怎麼試都差一點過 `strong`，`private` 卻過了 `strong`。