

Semi-Supervised Learning

B092040016 陳昱逢

Assignment 3

1 模型架構設計

現今有許多卷積神經網路 (convolution neural network; CNN) 架構，他們一開始多用較大的卷積核 (kernel size) 先去取得初步特徵，但有論文提出通過堆疊多層 3×3 的卷積核也可以達到使用較大的卷積核獲得的大範圍接受域 (receptive field)。因此本架構的發想都是基於多層 3×3 的卷積核堆疊而成，其中 conv4 接收的輸入除了來自 conv3 的輸出之外，還會加上 conv2 的輸出，同理，conv5 輸入來自 conv4 的輸出以及 conv3 的輸出，此想法來自於 ResNet [1] 提出的 residual 概念。Table 1 呈現我發想的模型架構。

Table 1: Architectures of my CNN model

variable name	convolution architecture
conv1	$3 \times 3, 64$ $3 \times 3, 128$ $3 \times 3, 194$
Maxpool	$kernel\ size = 2$ $stride = 2$
conv2	$3 \times 3, 256$
conv3	$3 \times 3, 512$
conv4	$3 \times 3, 1024$
conv5	$3 \times 3, 2048$
Maxpool	$kernel\ size = 2$ $stride = 2$
FC	5-layer FC

2 實驗

本實驗採用自訓練 (self-training) 的方式來完成半監督學習 (semi-supervised learning) 的任務，此部分將分為以下兩步驟去討論：第 2.1 節利用有標記的資料訓練和第 2.2 節利用沒有標記的資料產生偽標記去訓練

2.1 利用有標記的資料訓練

此步驟我資料擴增的方法採取了水平垂直翻轉、旋轉以及縮放等操作再進行訓練。在訓練時，我採取了 AdamW 作為我的 optimizer，損失函數的評估方式為 Cross-entropy，一開始我是使用 RMSProp 作為我的 optimizer，但後來經過多次測試，從 RMSProp 換到 Adam 再換到目前所用的 AdamW，收斂的速度都有依序變快，原因可能為 AdamW 考慮了 momentum 以及 L2 正則化，可以提升模型的收斂速度以及泛化能力。此外，我也有採用了 ReduceLROnPlateau 的學習速率調整策略，他會根據我驗證集的準確率指標來調整學習速率，當我的驗證集的準確率指標停滯不前時就會降低學習速率。Table 2 呈現了我訓練時 optimizer 以及 learning rate scheduler 的參數

Table 2: Configurations of optimizer and learning rate scheduler

Methods	Configurations
AdamW	$learning_rate = 0.001$ $weight_decay = 0.01$
ReduceLROnPlateau	$mode = 'max'$ $factor = 0.1$ $patience = 30$

2.2 利用沒有標記的資料產生偽標記去訓練

此步驟主要想利用沒有被標記的資料提升模型的表現度，透過前一步驟訓練好的模型預測沒有標記的資料，當資料通過 softmax 的幾率或信心程度超過一個臨界值，我們就給他一個偽標記，並把他加入訓練資料裡進行訓練，不斷地重複預測、拿到偽標籤、訓練等步驟。跟前一步驟一樣，我採取了 AdamW 作為我的 optimizer，cross-entropy 作為我的損失函數，我同樣也採用了 ReduceLROnPlateau 的學習速率調整策略，根據我驗證集的表現進而調整學習速率。Table 3 同樣呈現了我在此階段訓練時 optimizer 以及 learning rate scheduler 的參數。

對於臨界值的部分，我採取的是動態調整臨界值的方式，主要想法為當當前 epoch 驗證集的準確度超過過去之前所有 epoch 的驗證集最高準確度的話，我們就將臨界值調低，好讓更多未被標記的圖片可以成功被標記進而加進訓練資料訓練，反之則調高。臨界值 γ 的調整公式如下：

$$\gamma_{t+1} = \begin{cases} \max(0.6, \gamma_t \times 0.95), & \text{if } valid_acc > best_acc \\ \min(0.8, \gamma_t \times 1.05), & \text{otherwise,} \end{cases} \quad (1)$$

其中 t 為第 t 個 epoch， $valid_acc$ 表示當前 epoch 驗證集的準確度， $best_acc$ 則代表過去之前所有 epoch 的驗證集最高準確度。

Table 3: Configurations of optimizer and learning rate scheduler

Methods	Configurations
AdamW	$learning_rate = 0.001$ $weight_decay = 0.01$
ReduceLROnPlateau	$mode = 'max'$ $factor = 0.1$ $patience = 7$ $cooldown = 5$ $min_lr = 10^{-15}$

3 結論與發現

以下是我針對本次實驗的心得整理以及發現：

1. RMSProp 相較 Adam 收斂速度較慢。
2. 剛開始使用 Adam 時學習率調太高 (0.01 到 0.1 之間) 導致訓練結果較不佳。
3. 在參考老師的講義後，發覺 Adam 的學習率不能調太高。
4. 在試驗幾次後，決定以 AdamW 有考量 L2 正則化的情況下搭配 ReduceLROnPlateau 的學習率調整策略作為我最終的訓練方法可以達到較好的結果。
5. 使用動態調整臨界值時，相比只有訓練有標記的資料提升了 2% - 3% 的準確率。

此外，我有實作不同 CNN 的模型，分別為 ResNet [1] 以及 Inception-ResNet-v2 [2]，這裡註記在比較時沒有使用他們 pretrain 過後的模型或是權重去做比較，這裡發現使用大模型的架構時，可能反而表現較差，可能因為此次作業的資料為較小的資料集，因此可能在深的模型上表現較差。Table 4 展示我的模型與其他模型的實驗數據比較。

Table 4: Simulation results of each model

Model	Accuracy
ResNet18	0.7582
Inception-ResNet-v2	0.7910
MyModel	0.7817

References

- [1] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015.
- [2] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” 2016.