

機器學習 HW3 bonus: Deep Q

電機四 B04505025 陳在賢

一、做了那些比較的實驗

我有調的參數包含以下：

- (1) seed: 設定產生隨機數的 random seed，原本為 None，但為了提升模型的再現性 (reproduce)，所以將其設為正整數。另外還能搭配些運氣，產生能 train 出更佳模型的隨機數
- (2) “pretrained”: 設定是否要 pretrain model，原本設定為 False。但經 pretrain 後，準確率會大大提升，故恆改為 True。
- (3) torch.optim.Adam(params, lr=0.001): 設定 optimizer 的種類及其 learning rate，原先使用 SGD，我另外有用 Adam 實作，效能差異留至第二題討論。
- (4) “batchsize”: 指定一個 batch 中的 data 數，原本設定為 32。我將其降為 16，雖然會增加 training time，但較能避免掉入 local minimum。
- (5) “drop_out”: 調整 drop out 節點數比例，原本設定 None，我在實作時有開啟此功能，效能差異一樣留至第二題討論。
- (6) n_epoch == ...: 設定執行多少個 epoch 後，將模型參數輸出。由於我有設定 random seed，能確保每次 training，模型的變化過程類似。因此每次調完參數後，我都會先完整 train 完 50 個 epoch，同時觀察 epoch 為多少時會 train 出最佳模型，並將 n_epoch 設定好後再重新 train。

二、最後選擇此值 / 此 scheduling 的原因

這裡針對 optimizer, learning rate, drop out, n_epoch 四參數的選擇進一步說明。

起初把 pretrain 開啟後直接 train，validation accuracy 容易有不穩定的大起伏，有時相連兩 epoch 一次高至 0.90、一次低至 0.82。因此先將 learning rate 調小，限制模型的更新幅度。但仍無法完全解決此問題，後來猜測是因為 SGD 也有隨機性，因此改為用 Adam optimizer，之後 validation acc. 就比較沒有不穩定跳動。

但又發現：即便 train 到後面 training acc. 可高達 0.999，但 validation acc. 仍只有大概 0.86。後來發現 training data 只有 2500 筆，複雜的模型很容易發生 overfitting！因此先開啟 drop out 功能，再次限縮每次更新模型時的改變量。改完後可明顯發現起初幾個 epoch，validation acc. 已較 training acc. 高，因此提升後者的同時，前者也不斷提升，最高可到 0.9。

不過當 validation acc. 過了極值後往往會開始下降，而這很明顯又發生 overfitting 了，因此還需設定 n_epoch 以提早在 overfitting 發生前就存取模型參數。這也是為什麼我的 n_epoch 只設定為 15~20，一來可避免 overfitting，二來也能明顯降低 training time。