

目錄

★第一篇 PyTorch 的核心

第 1 章：深度學習與 PyTorch 函式庫

- 1-1 深度學習的演變
- 1-2 將 PyTorch 應用到深度學習中
- 1-3 為什麼要使用 PyTorch？
- 1-4 PyTorch 如何協助深度學習專案的開發
- 1-5 軟硬體需求

第 2 章：預先訓練的模型

- 2-1 利用預先訓練的模型來辨識圖像中的物體
- 2-2 利用預先訓練的模型來生成『假』圖片
- 2-3 能描述場景的神經網路模型
- 2-4 Torch Hub
- 2-5 結論

第 3 章：介紹張量

- 3-1 浮點數的世界
- 3-2 張量：多維陣列
- 3-3 利用索引值操作張量
- 3-4 為張量命名
- 3-5 張量的元素型別
- 3-6 其他常用的張量功能
- 3-7 張量的儲存原理
- 3-8 大小、偏移及步長
- 3-9 把張量移到 GPU 上
- 3-10 與 NumPy 的互通性
- 3-11 通用的張量（可適用於各種硬體）
- 3-12 將張量序列化（長期儲存）
- 3-13 結論

第 4 章：用張量表示現實中的資料

- 4-1 圖片資料
- 4-2 3D 立體圖片資料
- 4-3 表格資料
- 4-4 時間序列資料
- 4-5 表示文字資料
- 4-6 結論

第 5 章：學習的機制

- 5-1 學習的流程
- 5-2 學習就是在估算參數
- 5-3 以降低損失為目標

- 5-4 梯度下降演算法
- 5-5 用 PyTorch 的 autograd 進行反向傳播
- 5-6 結論

第 6 章：使用神經網路來擬合資料

- 6-1 神經元
- 6-2 PyTorch 的 nn 模組
- 6-3 進入正題：神經網路
- 6-4 結論

第 7 章：從圖片中學習

- 7-1 由小型圖片組成的資料集
- 7-2 區分鳥和飛機
- 7-3 結論

第 8 章：卷積神經網路

- 8-1 關於卷積
- 8-2 卷積的實際應用
- 8-3 建立 nn.Module 的子類別
- 8-4 訓練卷積網路
- 8-5 模型設計的進階技巧
- 8-6 結論

★第 2 篇 從現實世界中的圖片學習：肺部腫瘤偵測專案

第 9 章：利用 PyTorch 對抗癌症

- 9-1 應用實例的介紹
- 9-2 為大型專案做準備
- 9-3 CT 掃描到底是什麼？
- 9-4 專案概述：肺部腫瘤偵測器
- 9-5 結論

第 10 章：匯入原始資料並整合為資料集

- 10-1 原始 CT 資料
- 10-2 分析 LUNA 的標註資料
- 10-3 匯入 CT 掃描資料
- 10-4 定位結節
- 10-5 簡單的資料集實作
- 10-6 結論

第 11 章：訓練模型分辨結節的真假

- 11-1 最基礎的模型與訓練迴圈
- 11-2 程式的進入點
- 11-3 訓練前的設定與初始化

- 11-4 我們的首個神經網路
- 11-5 模型的訓練與驗證
- 11-6 輸出表現評估資料
- 11-7 執行訓練程式
- 11-8 評估模型表現：得到 99-7%的分類準確率就代表結束了嗎？
- 11-9 使用 TensorBoard 畫出訓練評估指標
- 11-10 為什麼模型沒有學會偵測結節？
- 11-11 結論

第 12 章：利用評估指標和資料擴增來改善訓練成效

- 12-1 模型改善的大方向
- 12-2 偽陽性與偽陰性
- 12-3 陽性與陰性的視覺化描述
- 12-4 理想的資料集長什麼樣子？
- 12-5 過度配適
- 12-6 利用資料擴增來防止過度配適
- 12-7 結論

第 13 章：利用『分割』找出疑似結節的組織

- 13-1 在專案中加入第二個模型
- 13-2 不同類型的資料分割
- 13-3 語義分割：像素層級的分類
- 13-4 更改模型以執行資料分割
- 13-5 更改資料集以執行資料分割
- 13-6 更改訓練程式以執行資料分割
- 13-7 最終結果
- 13-8 結論

第 14 章：端到端結節偵測（與未來方向）

- 14-1 迎向終點線
- 14-2 保持驗證資料集的獨立性
- 14-3 連接『分割模型』與『分類模型』
- 14-4 量化驗證
- 14-5 預測惡性結節
- 14-6 利用模型進行診斷
- 14-7 補充資源與資料
- 14-8 結論