深度學習

PyTorch 作業

繳交日期：2026年1月12日 23:59 (星期一);不能遲交

自主學習週加分題（學期總成績額外加 8%）

助教PyTorch輔導課: 2026年1月2日(星期五) 18:40開始於SF645現場授課;
同時也會開放Teams線上直播，課後TronClass公佈教材與錄影檔。

1. 目的

利用 PyTorch 實作具備現代化架構的卷積神經網路 (Modern CNN)

- 熟悉 PyTorch 框架：包含 Dataset、DataLoader 與 CNN 模型建置流程。
- 從 Layer 到 Block：掌握現代 Deep Learning 關鍵技術，利用殘差連接(Residual Connection) 與批次正規化(Batch Normalization; BN) 實作Residual Block，解決深層網路難以訓練的問題。
- 對抗過擬合 (Overfitting)：組合資料增強 (Data Augmentation) 與 Dropout 的正則化技術 (Regularization)，並分析其效果。
- 實驗比較不同架構（有無 Residual Block/ Data Augmentation + Dropout），對模型效能的影響。

2. 作業說明與整體流程

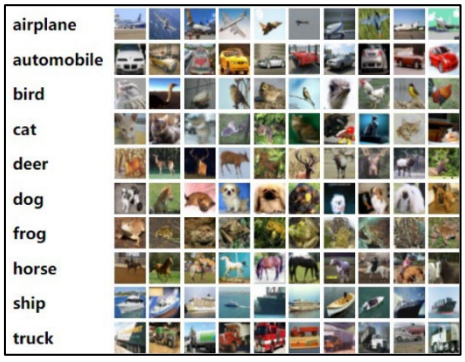
1. 準備資料集與 DataLoader
2. 實作 Residual Block
3. 建立 CNN 模型
4. 進行三組對照實驗
5. 分析與比較實驗結果

3. 實驗內容

3.1 步驟 1：資料集/資料前處理/資料增強

3.1.1 資料集

本作業採用CIFAR10數據集，原始資料集包含 10 個類別，每筆資料為 32×32 彩色圖像，共有50,000筆訓練資料和10,000筆測試資料。



- 為了聚焦於模型架構比較，本作業僅採用部分 CIFAR10 資料，並僅使用以下 5 個類別：airplane、automobile、cat、dog、ship。
- 數據設定：
  - 訓練與驗證資料共 10,000 筆，用於模型訓練與調整，建議切分 80% 訓練、20% 驗證。
  - 測試資料 2,000 筆（無類別標籤），用於最終預測。

3.1.2 資料前處理

本作業的影像前處理包含以下兩個概念步驟，實作時可於同一個 transformation pipeline 中完成：

- 將影像轉換為 Tensor，並將原始像素值從 [0, 255] 縮放至 [0, 1] 範圍。
- 對像素值進行正規化（Normalization），將其映射至 [−1, 1] 範圍，以提升模型訓練的效率與穩定性。建議使用以下正規化參數：mean = 0.5，std = 0.5。

3.1.3 資料增強

為了提升模型泛化能力，請在訓練集加入至少兩種增強技術，例如：

- RandomHorizontalFlip (隨機水平翻轉)
- RandomCrop (隨機裁減，需搭配 Padding)
- RandomRotation(degrees = 15) (隨機旋轉)
- ColorJitter(brightness = 0.2, contrast = 0.2) (色彩抖動)

(注意：驗證集與測試集，僅需資料前處理，不可加入資料增強處理。)

3.2 步驟 2：設計 Building Blocks

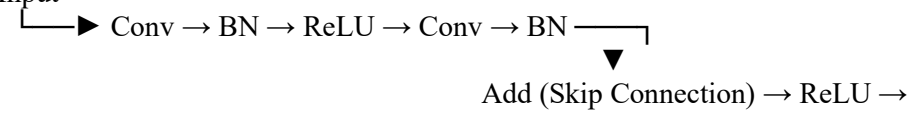
本作業要求設計兩種基礎區塊，以進行公平的對照實驗。

3.2.1 Plain Block (用於 Baseline)

- 注意：無 Batch Normalization，無 Shortcut。
- 範例結構：Conv3 × 3 → ReLU → Conv3 × 3 → ReLU

3.2.2 Residual Block (用於 Model A & B)

- 注意：需實作 Shortcut 加法與 Batch Normalization。
- 範例結構：



- 範例說明：包含兩層卷積層，每個卷積層後需搭配 Batch Normalization，殘差路徑（Shortcut） 需與主路徑相加 （Element-wise Addition），最後再通過非線性函數（ReLU）。
- Residual Block 實作結構說明：
  - 可自行設計「Convolution + BN + ReLU」的次數，及濾波器大小 (Filter Size) 。
  - 注意：本作業實作 Identity Shortcut: $\mathbf{y} = F(\mathbf{x}) + \mathbf{x}$，輸入與輸出特徵圖形保持一致，Conv (stride = 1, padding = same)，確保輸入輸出維度相同，無需實作 Projection Shortcut。）
- Residual Block 實作說明：在現代深度學習實務中，通常將「Convolution + BN + ReLU」視為一個基本運算單元（Building Block），而 Residual Block 則是由多個此類運算單元，搭配捷徑連接（Skip Connection）所組成。透過 Residual Connection，模型能有效緩解深層網路訓練時的梯度消失問題，使網路在加深後仍能穩定收斂。

3.3 步驟 3：實驗設計與模型比較

為確保三組實驗具有可比性，以下提供一個整體CNN模型架構範例作為參考：

- 輸入層：影像大小 3 × 32 × 32
- 前段：一層 Conv (將 Input 轉為 Feature Map)。
- 中段：堆疊 2~3 個 Block (Baseline 用 Plain Block；Model A/B 用 Residual Block)。
  - 說明： Plain Block 與 Residual Block 內的主路徑完全相同 (Conv → ReLU → Conv → ReLU...)，但 Plain Block 移除 Shortcut (加法操作) 與 BN。
- 後段(分類器)：全局池化層 (Global Average Pooling; GAP) 縮減為 1 × 1 → Flatten → Dropout (僅 Model B) → Linear。
  - 說明：若中段未進行空間降維，而且本作業要求實作 Residual Block 時，採用 Identity Shortcut 。假設最後一層 Residual Block 輸出通道數 (Channels) 為 64，則全連接層 (Flatten) 的向量長度為：64 × 32 × 32 = 65,536。而全連接層輸出為 5 類，因此參數權重為 5 × 65,536 ≈ 32萬個參數。參數量非常龐大，容易導致

Overfitting 或記憶體不足。因此建議使用 GAP 把整張特徵圖平均成一個點，再進行展平 (Flatten)。

- 實作提示：可以使用 nn.AdaptiveAvgPool2d((1, 1)) 接著 torch.flatten() 或 nn.Flatten()。

- 輸出層：5 類別（使用 CrossEntropyLoss，模型中不需顯式加 Softmax）

注意：

1. CNN模型深度不需過深，重點在於正確實作 Residual Block，並能穩定完成訓練。
2. 本作業重點不在於 Max Pooling 或分類層的設計細節，而在於Residual Block與整體訓練流程的正確性。因此可以採用或不採用Max Pooling 以降低特徵圖維度，而分類層亦可使用若干或單一Fully Connected Layer。

為了系統性分析 Residual Block 與 Data Augmentation + Dropout 對模型效能的影響，請在相同訓練設定下，完成以下三組實驗：

(實驗一) Baseline (同深度的 Plain CNN 模型）

- 模型：一般 CNN，沒有 Residual Block，沒有 Data Augmentation + Dropout；必須與 Model A 擁有相同的卷積層數量與通道數；若採用 Max Pooling 或有若干 Fully Connected Layer，必須與 Model A 架構一致。
- 目的：作為對照組，觀察在相同深度下，缺乏現代機制時的訓練狀況，以突顯 Residual Block 與 Data Augmentation + Dropout 的實際貢獻。

(實驗二) Model A：(具備 Residual Block 的 CNN)

- 模型：具有 Residual Block，沒有 Data Augmentation + Dropout。必須與 Baseline 的卷積層數量與通道數相同；若採用 Max Pooling 或有若干 Fully Connected Layer，必須與 Baseline 架構一致。
- 目的:驗證 Residual Block 是否加速收斂,並解決深度網路難以訓練的問題。

(實驗三) Model B（完整現代化模型）

- 模型：架構同 Model A，訓練時啟用 Data Augmentation，並於分類器加入 Dropout (建議 $p = 0.3 \sim 0.5$)。
  - 注意：Dropout 僅於後段(分類器) 的 Model B 啟用，不適用 Model A中段的 Residual Block ，以避免干擾 Residual Block 本身對收斂行為的影響。
- 目的：驗證正則化技術對抗 Overfitting 的能力，即Data Augmentation + Dropout 是否能進一步提升模型的泛化能力。

3.4 訓練設定要求（適用於三組實驗）

- Loss Function：CrossEntropyLoss
- Optimizer：SGD 或 Adam
- Epoch：建議 20–50
- Batch Size：建議 64 或 128
- 建議使用 Early Stopping 避免 Overfitting
- 請記錄並保存每個 epoch 的：
  - Training Loss
  - Validation Loss
  - Training Accuracy
  - Validation Accuracy

4. 結果報告

請撰寫一份完整的實驗結果報告，內容包含文字說明與圖表分析，至少涵蓋以下項目：

(一) 模型架構與訓練行為分析

- 說明模型架構設計
- 繪製三組模型：
  - Training / Validation Loss 曲線
  - Training / Validation Accuracy 曲線
- 比較不同模型的收斂速度與穩定性

(二) 模型效能比較

以表格整理三組模型在驗證集上的最終準確率，並觀察：

- Residual Block 是否改善模型訓練的收斂速度與穩定性
- Data Augmentation + Dropout 是否降低過擬合現象

(三) 其它心得與討論

其他觀察與發現，或對學習過程與結果進行反思。

5. 評分標準

- 程式碼正確性與完整性 (50%)：
  - 程式碼可讀性與註解
  - Baseline 模型實作
  - 正確實作 Residual Block
  - 正確實作 Data Augmentation + Dropout
  - 程式可執行並產出結果
- 結果報告與分析 (40%)：
  - 模型架構設計說明清楚
  - 三組比較實驗的數據與圖表呈現
  - 對實驗結果有合理的邏輯分析
  - 心得與討論有獨到的見解
- 測試準確率 (10%)：
  - Model B (完整現代化模型) 在測試集上的表現。