CNN 模型來源 HUNG-YI LEE (李宏毅) MACHINE LEARNING
2023 SPRING: https://speech.ee.ntu.edu.tw/~hylee/ml/2023-spring.php

主題: CNN-影像分類

主題描述:
使用卷積神經網路 (CNN) 來解決影像分類問題，完成數據增強 (Data Augmentation)、CNN 模型訓練和預測結果。

引用的資料與程式碼敘述:

引用的資料:
Dataset 使用 food-11 資料集，該資料集包含 11 種不同類別的食物影像。

- 訓練集：10,000 張標記影像
- 驗證集：3,643 張標記影像
- 測試集：3,000 張未標記影像
- Link:https://www.dropbox.com/scl/fi/w5r49vs5e956w6c4z6ob9/food-11.zip?rlkey=3no5l2xjiqgk2ckwbewaanm5p&e=1&dl=0

程式碼敘述:
1. 用 `nvidia-smi`，用戶可以方便地監控和管理 GPU 資源。

2. 通過 Dropbox 或 Google Drive 下載 `food-11.zip` 文件並保存 `food11.zip`資料集，下載到本地系統後解壓。

3. import 需要的庫，設置 PyTorch 環境，設置隨機種子確保結果的可重複性。

4. 定義兩個圖像（`transforms`），分別用於測試和訓練階段，進行圖像預處理；定義自定義數據集類 `FoodDataset`，用於加載食物圖像數據集。

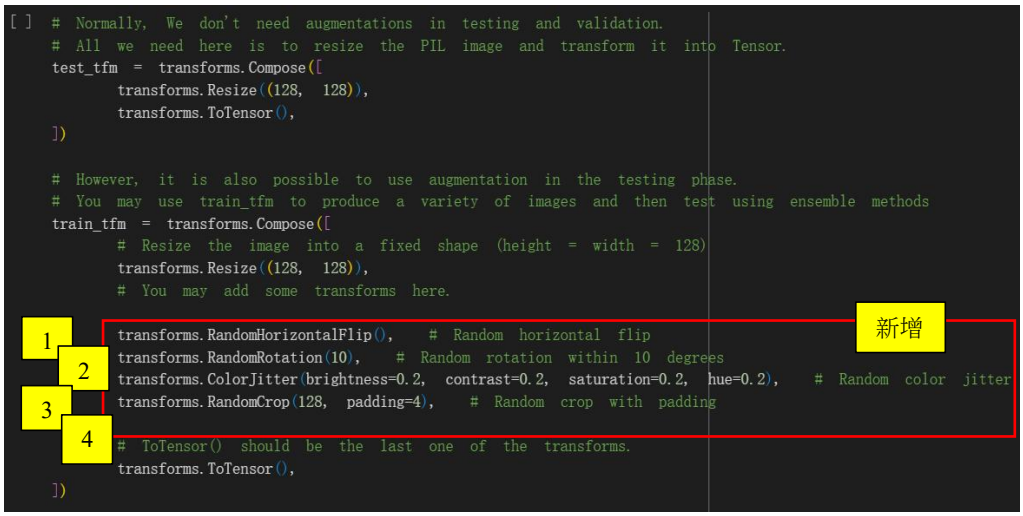
5. 定義 CNN 結構模型。

6. 設定裝置、初始化模型、設定超參數、定義損失函數和優化器；載入資料進行訓練和驗證來完成影像分類模型的訓練。

7. 訓練過程中若模型性能提升則保存最佳模型。

自行修改的部分比對:

新增以下
1.transforms.RandomHorizontalFlip()
2.transforms.RandomRotation(10)
3.transforms.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue=0.2)
4.transforms.RandomCrop(128, padding=4)



1. transforms.RandomHorizontalFlip():
- 默認情況下，從左到右以 0.5 的概率隨機水平翻轉圖片。
- 效果為幫助模型學習不變特徵，通過觀察圖片的原始版本和翻轉版本來增加模型的魯棒性。這樣可以提高模型對不同物體方向變化的穩定性。

2. transforms.RandomRotation(10)：
- 會在 -10 到 10 度範圍內隨機旋轉圖片。
- 效果為有助於模型對小角度旋轉具有不變性，這在實際場景中非常有用，因為圖片中的物體不總是完美對齊。

3. transforms.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue=0.2)：
- 會隨機改變圖片的亮度、對比度、飽和度和色調。
- 效果為它模擬了不同的照明條件和顏色變化，使模型在遇到這些情況時更加穩定。
- 參數（亮度、對比度、飽和度、色調）都指定了一個範圍。`brightness=0.2` 意味著亮度可以在 `0.8`（1 - 0.2）和 `1.2`（1 + 0.2）之間調整。

4. transforms.RandomCrop(128, padding=4)：
- 會隨機裁剪圖片到指定大小（128x128 像素），並在裁剪之前給圖片每邊添加 4 像素的填充。
- 效果為它幫助模型學習在物體部分可見或位於不同位置時仍能識別它們。
- 填充添加 4 像素的填充確保隨機裁剪不總是集中在圖片的中心部分，從而增強訓練樣本的多樣性。

5.將 n_epochs = 8 改為 n_epochs = 20

```
[ ]  # "cuda" only when GPUs are available.
     device = "cuda" if torch.cuda.is_available() else "cpu"

     # Initialize a model, and put it on the device specified.
     model = Classifier().to(device)

     # The number of batch size.
     batch_size = 64

     # The number of training epochs.
5    n_epochs = 20

     # If no improvement in 'patience' epochs, early stop.
     patience = 5

     # For the classification task, we use cross-entropy as the measurement of performance.
     criterion = nn.CrossEntropyLoss()

     # Initialize optimizer, you may fine-tune some hyperparameters such as learning rate on your own.
     optimizer = torch.optim.Adam(model.parameters(), lr=0.0003, weight_decay=1e-5)
```

5.將 n_epochs = 8 改為 n_epochs = 20

- 可增加模型學習時間：讓模型有更多時間學習數據特徵和模式。
- 逐漸收斂：深度學習模型通常需要多個訓練週期才能收斂到最佳狀態，將 n_epochs 增加到 20 可以幫助模型更接近最佳性能。

原本模型(acc = 0.6180)

```
[ Train | 006/008 ] loss = 0.88748, acc = 0.69258
100% ████████████████████ 57/57 [00:18<00:00,  3.07it/s]
[ Valid | 006/008 ] loss = 1.14418, acc = 0.61604
[ Valid | 006/008 ] loss = 1.14418, acc = 0.61604 -> best
Best model found at epoch 5, saving model
100% ████████████████████ 157/157 [01:04<00:00,  2.88it/s]
[ Train | 007/008 ] loss = 0.76489, acc = 0.73846
100% ████████████████████ 57/57 [00:18<00:00,  2.76it/s]
[ Valid | 007/008 ] loss = 1.17393, acc = 0.61800
[ Valid | 007/008 ] loss = 1.17393, acc = 0.61800 -> best
Best model found at epoch 6, saving model
100% ████████████████████ 157/157 [01:05<00:00,  3.00it/s]
[ Train | 008/008 ] loss = 0.65931, acc = 0.77110
100% ████████████████████ 57/57 [00:18<00:00,  3.77it/s]
[ Valid | 008/008 ] loss = 1.31970, acc = 0.59155
[ Valid | 008/008 ] loss = 1.31970, acc = 0.59155
```

改動後(acc = 0.63470)

```
[ Train | 018/020 ] loss = 0.71189, acc = 0.74751
100% ████████████████████ 57/57 [00:19<00:00,  2.22it/s]
[ Valid | 018/020 ] loss = 1.19921, acc = 0.62237
[ Valid | 018/020 ] loss = 1.19921, acc = 0.62237
100% ████████████████████ 157/157 [01:33<00:00,  2.49it/s]
[ Train | 019/020 ] loss = 0.68887, acc = 0.76304
100% ████████████████████ 57/57 [00:18<00:00,  3.82it/s]
[ Valid | 019/020 ] loss = 1.29957, acc = 0.59004
[ Valid | 019/020 ] loss = 1.29957, acc = 0.59004
100% ████████████████████ 157/157 [01:33<00:00,  2.49it/s]
[ Train | 020/020 ] loss = 0.65087, acc = 0.77239
100% ████████████████████ 57/57 [00:19<00:00,  3.04it/s]
[ Valid | 020/020 ] loss = 1.12264, acc = 0.63470
[ Valid | 020/020 ] loss = 1.12264, acc = 0.63470 -> best
Best model found at epoch 19, saving model
```

心得敘述:

通過數據增強，像是隨機翻轉、旋轉、色彩抖動和隨機裁剪，並將訓練週期增加到 20，模型的準確率從 0.6180 提升到 0.6347。看到這些改動後模型性能提升，讓我感到很有成就感。