

國立成功大學
數據科學研究所
自動駕駛實務

Project 2 : Traffic Sign Classifier

Traffic Sign Classifier using LeNet

研究生：曾文海
指導教授：莊智清 傅志雄

中華民國一一三年五月

目錄

目錄	1
第一章 緒論.....	1
1-1. 研究背景	1
第二章 模型架構.....	2
2-1. LeNet網路架構	2
第三章 實驗結果.....	4
3-1. 結果可視化	4
第四章 結果及討論.....	7
4-1. 結論與討論	7

第一章 緒論

1-1. 研究背景

道路基礎設施的演進歷史密切關聯著人類視覺系統的發展。人類視覺系統通過感知和控制，在環境與車輛之間傳遞重要資訊，是道路交通安全的關鍵。自動駕駛系統要安全地在現有道路環境中運行，至少需要具備與人類同等水準的視覺識別能力。在自動駕駛車輛的視覺感知中，正確分類交通標誌是至關重要的任務。

德國交通標誌識別基準（GTSRB）數據集是一個多類別、單一圖像分類挑戰。該數據集由德國道路上拍攝的交通標誌圖像組成，每個圖像都附有相應的類別標籤。這些圖像是從車載移動攝像頭的視頻數據中提取的，去除了所有時間資訊。

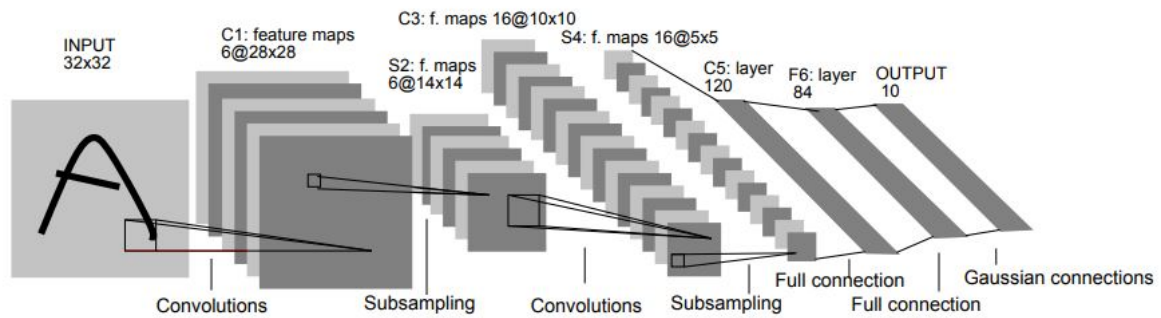
人類視覺系統的進化與道路基礎設施的發展息息相關。自動駕駛技術必須在視覺感知方面達到人類水準，而德國交通標誌識別基準數據集則提供了一個重要的測試平臺，用於評估自動駕駛系統對交通標誌的準確識別能力。



圖 1.1: 資料集類別共43類

第二章 模型架構

2-1. LeNet網路架構



LeNet 是由 Yann LeCun 團隊所開發的早期卷積神經網路架構，廣泛被認為是卷積神經網路的鼻祖。這一網路架構主要用於手寫數字圖像的辨識，其構造包括多個層次，分別是卷積層、池化層、全連接層，以及一個高斯連接層作為輸出層。

LeNet 的整體架構包含七個層次，分別為：第一個卷積層（C1）、第一個池化層（S2）、第二個卷積層（C3）、第二個池化層（S4）、全連接卷積層（C5）、全連接層（F6），以及最後的高斯連接層。其輸入層接受一個 28x28 的單通道影像。

在這個網路中，卷積層使用 5x5 的過濾器（Filter），其中第一個卷積層的輸出通道為 6，第二個為 16，並且兩個卷積層都使用 Sigmoid 函數作為激活函數。池化層採用 2x2 的窗口和步長（stride）為 2 的設定，並進行平均池化。

全連接層的結構包括兩個層，分別擁有 120 個和 84 個神經元。最終的輸出層，

即高斯連接層，使用徑向基函數（RBF，徑向歐式距離函數）來計算輸入向量和參數向量之間的歐式距離，用於生成 10 個輸出對應到 0 到 9 的手寫數字。

因此可以發現LeNet是一個既簡單又有效的卷積神經網路，接下來我希望使用pytorch套件來建構一個LeNet模型來訓練GTSRB資料夾的Traffic Sign Classifier任務。

第三章 實驗結果

3-1. 結果可視化

本章節將使用上第二章節所介紹的LeNet做影像分類，而要訓練的資料集是第一個章節介紹的GTSRB，交通號誌的影像分類在這個資料集中共有43資料集。本次訓練的epoch設定為20，損失函數使用Cross-entropy loss。

$$H(P^* | P) = - \sum_i \underbrace{P^*(i)}_{\text{TRUE CLASS DISTIRBUTION}} \log \underbrace{P(i)}_{\text{PREDICTED CLASS DISTIRBUTION}}$$

圖 3.1: Cross-entropy loss

Cross entropy loss（交叉熵損失）是一種常用於分類問題的損失函數。它衡量了兩個概率分佈之間的差異性，通常用於衡量模型預測的概率分佈與實際標籤之間的差異。

在機器學習中，當我們進行分類任務時，通常會將模型的輸出視為概率分佈，表示每個類別的可能性。而交叉熵損失則通過計算實際標籤對應的概率的負對數來評估模型的性能。當模型預測的概率分佈與實際標籤相符時，交叉熵損失會趨於最小值。

簡而言之，交叉熵損失可以幫助我們衡量模型預測的準確性，並通過最小化該損失函數來訓練模型，使其能夠更好地適應訓練數據並提高分類性能。

```
[Epoch 1/20] Train loss: 2.16 Validation loss: 0.92 Validation accuracy: 71.97%
[Epoch 2/20] Train loss: 0.63 Validation loss: 0.48 Validation accuracy: 85.84%
[Epoch 3/20] Train loss: 0.37 Validation loss: 0.32 Validation accuracy: 91.28%
[Epoch 4/20] Train loss: 0.27 Validation loss: 0.24 Validation accuracy: 93.51%
[Epoch 5/20] Train loss: 0.20 Validation loss: 0.18 Validation accuracy: 95.37%
[Epoch 6/20] Train loss: 0.16 Validation loss: 0.17 Validation accuracy: 95.50%
[Epoch 7/20] Train loss: 0.12 Validation loss: 0.15 Validation accuracy: 96.30%
[Epoch 8/20] Train loss: 0.11 Validation loss: 0.13 Validation accuracy: 96.71%
[Epoch 9/20] Train loss: 0.09 Validation loss: 0.13 Validation accuracy: 96.74%
[Epoch 10/20] Train loss: 0.07 Validation loss: 0.11 Validation accuracy: 97.21%
[Epoch 11/20] Train loss: 0.06 Validation loss: 0.11 Validation accuracy: 97.69%
[Epoch 12/20] Train loss: 0.06 Validation loss: 0.11 Validation accuracy: 97.39%
[Epoch 13/20] Train loss: 0.05 Validation loss: 0.12 Validation accuracy: 97.50%
[Epoch 14/20] Train loss: 0.05 Validation loss: 0.11 Validation accuracy: 97.35%
[Epoch 15/20] Train loss: 0.04 Validation loss: 0.15 Validation accuracy: 96.89%
[Epoch 16/20] Train loss: 0.04 Validation loss: 0.10 Validation accuracy: 97.83%
[Epoch 17/20] Train loss: 0.03 Validation loss: 0.13 Validation accuracy: 97.54%
[Epoch 18/20] Train loss: 0.03 Validation loss: 0.11 Validation accuracy: 97.76%
[Epoch 19/20] Train loss: 0.03 Validation loss: 0.21 Validation accuracy: 95.28%
[Epoch 20/20] Train loss: 0.02 Validation loss: 0.09 Validation accuracy: 98.34%
Test loss: 0.80 Test accuracy: 91.01%
```

圖 3.2: LeNet訓練了20 epoch

我將原本的驗證資料集取出20%做為測試集，在訓練後的驗證集準確度達到98.34%，在測試集的準確度達到91.01%，觀察驗證集的loss應該是模型還能有更好的表現，還可以調高Epoch來獲得更好的表現。

混淆矩陣是用於評估分類模型性能的一種矩陣。它以實際標籤和模型預測的類別為基礎，將測試集中的樣本按照它們的預測結果進行分類。混淆矩陣的行代表了實際標籤的類別，列則代表了模型預測的類別。在混淆矩陣中，每個元素表示實際屬於該行類別但被模型錯誤預測為該列類別的樣本數量。

混淆矩陣可以提供豐富的模型性能信息，例如模型的準確率、精確率、召回率

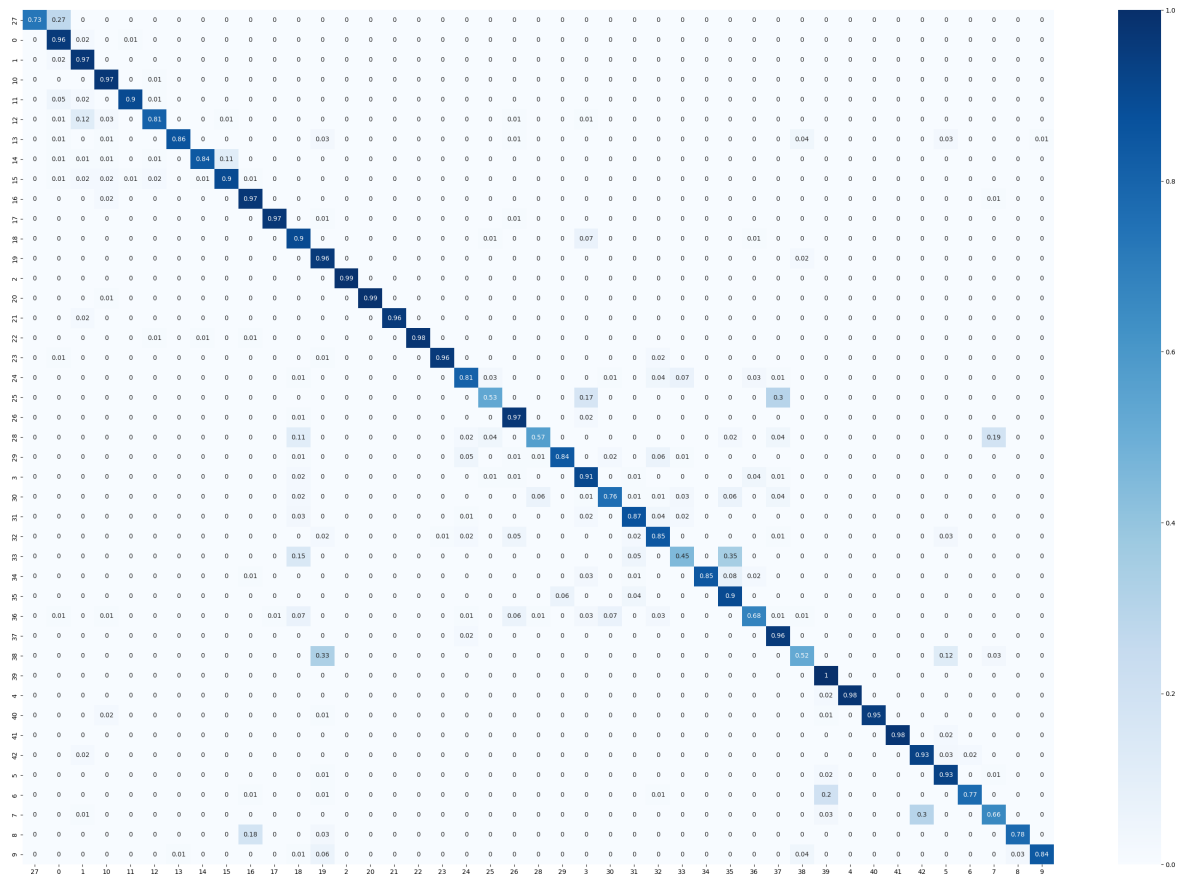


圖 3.3: 混淆矩陣

等。通過對混淆矩陣進行分析，我們可以了解模型在各個類別上的預測表現，進而針對性地進行模型調優和改進。混淆矩陣還可用於比較不同模型的性能，以幫助選擇最優的模型。

第四章 結果及討論

4-1. 結論與討論

透過深度神經網路LeNet做圖像分類任務，在影像處理上除了將影像RESIZE成32x32的大小外沒有做其他處理，本次訓練效果達到91.01%，除裡增加訓練的Epoch，還有增加影像處理的方式來增強模型表現，像是旋轉、水平翻轉、垂直翻轉等等。

The GitHub repository can be found at:

https://github.com/w5535586/2024_AD/tree/main/TrafficSign_Classifier_Project

參 考 文 獻

Lecun, Y. and Bottou, L. and Bengio, Y. and Haffner, P. (2011). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 2278-2324.