

Programming assignment

- **Classification**

- **資料處理**

首先使用 Python 的 `xml.etree.ElementTree` 解析 xml 檔案，提取經緯度。

再根據規定生成資料：

$$\text{label} = \begin{cases} 0, & \text{if 溫度為無效值}(-999.0) \\ 1, & \text{溫度為有效值} \end{cases}$$

最終得到(經度, 緯度, label)格式的資料，
接著為了更好的訓練均經過標準化處理。

- **切分資料**

以 7:1.5:1.5 的比例分成訓練集、驗證集、測試集。

- **模型架構**

使用 PyTorch 建立一個 Feedforward Neural Network。
輸入層: 2 個神經元，分別接收標準化後的經度與緯度
隱藏層 1: 16 個神經元，使用 'ReLU' 作為激活函數
隱藏層 2: 8 個神經元，使用 'ReLU' 作為激活函數
輸出層: 1 個神經元，使用 'Sigmoid' 激活函數

- **Loss function**

使用 Binary Cross-Entropy Loss

- **Optimizer**

使用 Adam

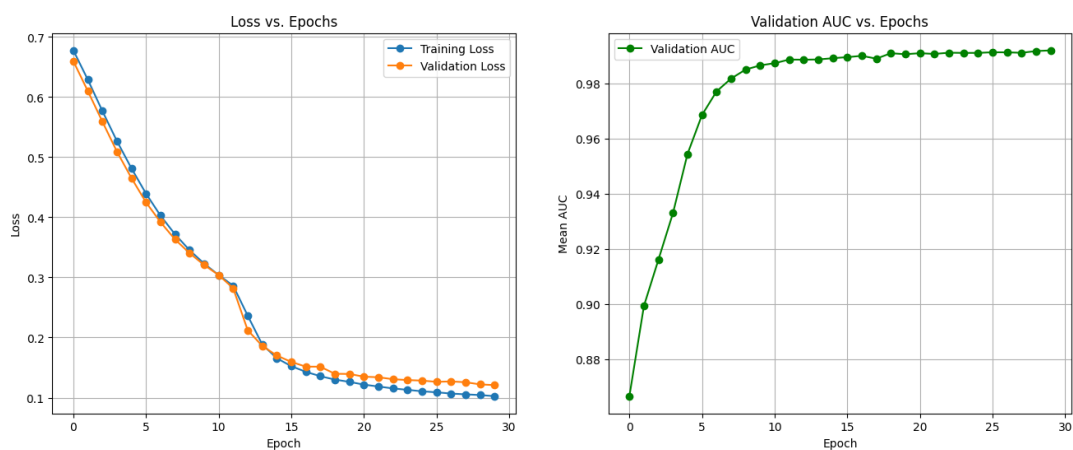
- **Configs**

Batch size : 64

Epochs : 30

Learning rate : 1e-3

■ 訓練圖



■ 結果

使用 AUC 這個指標來評估，測試資料對於模型的擬合情況

最終 Test Set AUC: 0.9941

從訓練的圖來看，訓練損失下降，驗證損失也跟著下降，沒有強烈的起伏或抖動，沒有 Overfitting 的問題，模型訓練的還行。

- regression

- 資料處理

首先使用 Python 的 `xml.etree.ElementTree` 解析 xml 檔案，提取經緯度。

再根據規定生成資料：

$$\text{label} = \begin{cases} 0, & \text{if 溫度為無效值}(-999.0) \\ 1, & \text{溫度為有效值} \end{cases}$$

再將攝氏溫度轉為華氏溫度，

最終產生（經度，緯度，華氏溫度）格式的資料，

接著為了更好的訓練均經過標準化處理。

- 切分資料

以 7:1.5:1.5 的比例分成訓練集、驗證集、測試集。

- 模型架構

使用 PyTorch 建立一個 Feedforward Neural Network。

輸入層: 2 個神經元，分別接收標準化後的經度與緯度

隱藏層 1: 32 個神經元，使用 'ReLU' 作為激活函數

隱藏層 2: 16 個神經元，使用 'ReLU' 作為激活函數

輸出層: 1 個神經元，不使用激活函數

- Loss function

使用 Mean Squared Error

- Optimizer

使用 Adam

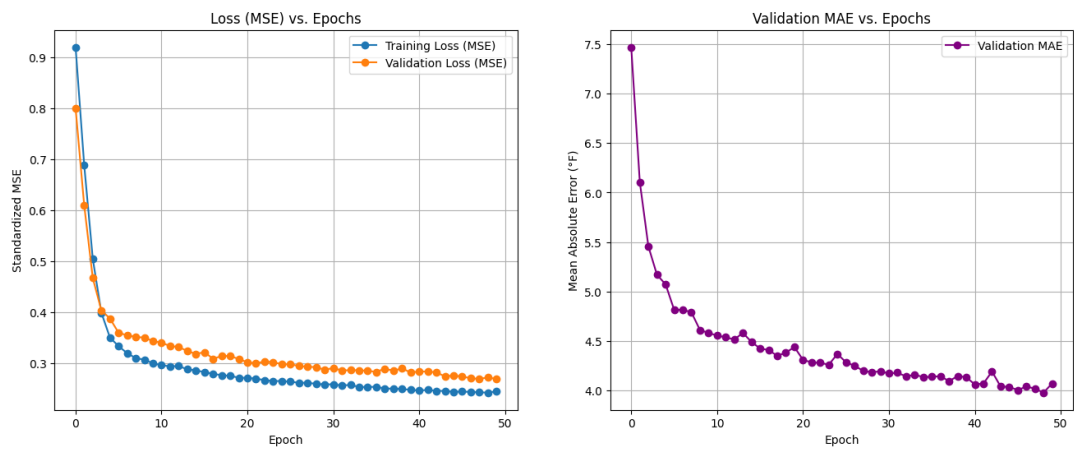
- Configs

Batch size : 32

Epochs : 50

Learning rate : 1e-3

■ 訓練圖



■ 結果

使用 MAE 這個指標來評估，測試資料對於模型的擬合情況

最終 Test Set MAE: 4.1809 °F

從訓練的圖來看，訓練損失下降，驗證損失也跟著下降，沒有強烈的起伏或抖動，沒有 Overfitting 的問題，模型訓練的還行。