

專案報告：基於神經網路的氣象格點資料分析(使用 Gemini 輔助)

姓名：陳昱瑜 學號：112613037 日期：2025 年 9 月 29 日

## 1. 摘要 (Abstract)

本報告旨在利用神經網路模型，對中央氣象署的「小時溫度觀測分析格點資料」進行分析。我們建立了兩種模型：(1) 一個用於預測溫度的回歸模型，以及 (2) 一個用於辨識陸地與海洋的分類模型。

實驗結果顯示，分類模型能以超過 97% 的準確率成功學習台灣的地理輪廓，但回歸模型的預測誤差 (MSE) 難以降低至 10 以下。經過深入分析，我們發現回歸模型性能的瓶頸主要源於輸入特徵的局限性，特別是缺少了關鍵的「海拔」特徵。本報告將詳細闡述資料處理流程、模型架構、訓練結果，並重點分析模型表現背後的數據洞見。

## 2. 前言與目標 (Introduction & Objectives)

### 2.1 資料集描述

簡要介紹使用的資料集 0-A0038-003.xml，說明它是一個 120x67 的網格資料，包含台灣地區的每小時溫度觀測值，無效值為 -999.0。

### 2.2 專案目標

目標一（回歸模型）：建立一個神經網路模型，根據經緯度座標預測對應的溫度。

目標二（分類模型）：建立一個神經網路模型，根據經緯度座標判斷該點為有效數據 1（陸地）或無效數據 0（海洋）。

## 3. 資料前處理 (Data Preprocessing)

### 3.1 資料讀取與轉換

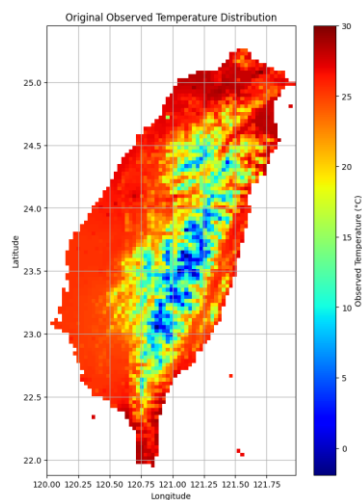
從原始類 XML 格式的文字檔中提取純數據字串，將字串清理、分割，並轉換為 NumPy 陣列，最終重塑為 (120, 67) 的溫度網格 temp\_grid。

### 3.2 訓練資料集建構

根據 temp\_grid 建構出兩個獨立的資料集：

data\_reg\_np：僅包含有效溫度點的 [經度, 緯度, 溫度] 列表。

data\_cla\_np：包含全部 8040 個 [經度, 緯度, 標籤] 列表（陸地為 1，海洋為 0）。



圖一，原始溫度分布圖

#### 4. 回歸模型：溫度預測

##### 4.1 模型架構

輸入層、3 層各含 64 個神經元及搭配 ReLU 激活函數的隱藏層、1 個輸出的線性單元。  $ReLU(x) = \max(0, x)$



--- 模型結構 ---

Model: "functional\_2"

| Layer (type)                   | Output Shape | Param # |
|--------------------------------|--------------|---------|
| input_coordinates (InputLayer) | (None, 2)    | 0       |
| hidden_layer_1 (Dense)         | (None, 64)   | 192     |
| hidden_layer_2 (Dense)         | (None, 64)   | 4,160   |
| hidden_layer_3 (Dense)         | (None, 64)   | 4,160   |
| output_temperature (Dense)     | (None, 1)    | 65      |

Total params: 8,577 (33.50 KB)

Trainable params: 8,577 (33.50 KB)

Non-trainable params: 0 (0.00 B)

損失函數(Loss Function)為 MSE(mean\_squared\_error)，優化器為 SGD(Stochastic Gradient Descent)。

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

$$\theta_{new} = \theta_{old} - \eta \cdot \nabla J(\theta; x^{(i)}; y^{(i)})$$

$\theta_{new}$  是更新後的參數值。  $\theta_{old}$  是更新前的參數值。

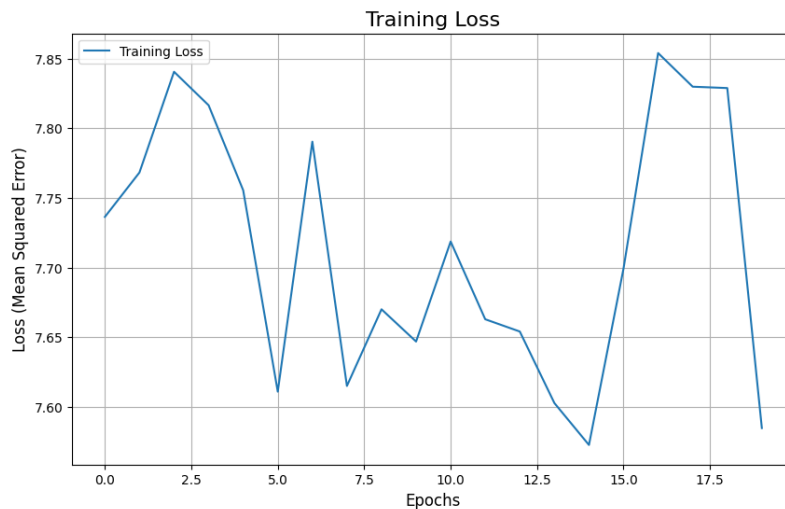
$\eta$  (Eta) 是學習率 (Learning Rate)，它控制了每一步走多大。這是一個需要手動設定的超參數。

$x^{(i)}; y^{(i)}$  是從訓練集中隨機選取的一個數據樣本及其對應的標籤。

$\nabla J(\theta; x^{(i)}; y^{(i)})$  是損失函數  $J$  對於參數  $\theta$  的梯度 (Gradient)，但這個梯度只根據單一的樣本  $(x^{(i)}, y^{(i)})$  來計算。它代表了在  $x^{(i)}$  這個點上，能讓 loss 下降最快的方向。

## 4.2 訓練過程與結果分析

發現一：高基準誤差



雖然模型有學習，但損失 loss 最終穩定在一個大於 7 的較高水平。

發現二：關鍵特徵的缺失——海拔

深入探討 loss 降不下去的根本原因可能是只用經緯度無法區分高山與平地，導致模型必然產生較大誤差。

## 5. 分類模型：陸地／海洋辨識

### 5.1 模型架構

輸入層、3 層各 64 個神經元及搭配 ReLU 激活函數的隱藏層、1 個輸出搭配

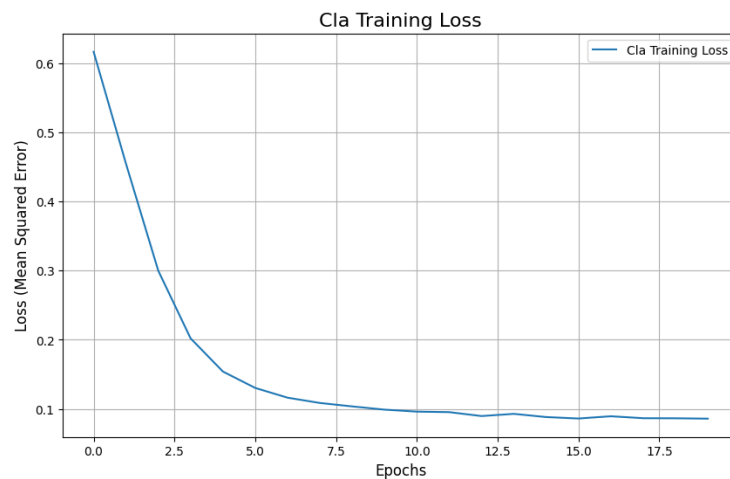
sigmoid 函數的線性單元。 $ReLU(x) = \max(0, x)$ ,  $\sigma(x) = \frac{1}{1+e^{-x}}$

--- 分類模型結構 ---  
Model: "functional\_3"

| Layer (type)                   | Output Shape | Param # |
|--------------------------------|--------------|---------|
| input_coordinates (InputLayer) | (None, 2)    | 0       |
| hidden_layer_1 (Dense)         | (None, 64)   | 192     |
| hidden_layer_2 (Dense)         | (None, 64)   | 4,160   |
| hidden_layer_3 (Dense)         | (None, 64)   | 4,160   |
| output_probability (Dense)     | (None, 1)    | 65      |

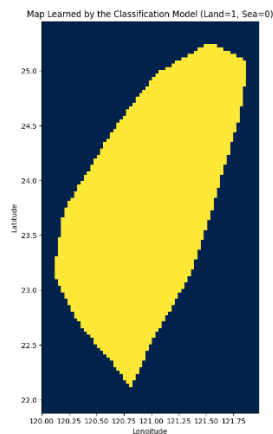
### 5.2 訓練過程與結果分析

發現一：loss 低



分析圖表，說明模型快速收斂，且達到了非常低的損失。

發現二：模型學到的地理輪廓



展示模型成功學習並繪製出台灣本島的輪廓，證明其有效的分類能力。

發現三：模型的泛化與簡化——消失的離島

模型忽略了澎湖、綠島等數據佔比極小的離島。

分析原因：模型為了追求整體損失最小化，選擇性地「犧牲」了這些會使決策邊界變得複雜的微小特徵。

## 6. 結論

### 6.1 總結

總結本次專案的成果：成功建立了高準確率的陸海分類模型，並透過對回歸模型的深入分析，揭示了特徵工程（Feature Engineering）對於模型性能的決定性影響。



### 6.2 程式碼: Weather\_2\_0.ipynb