Programming assignment

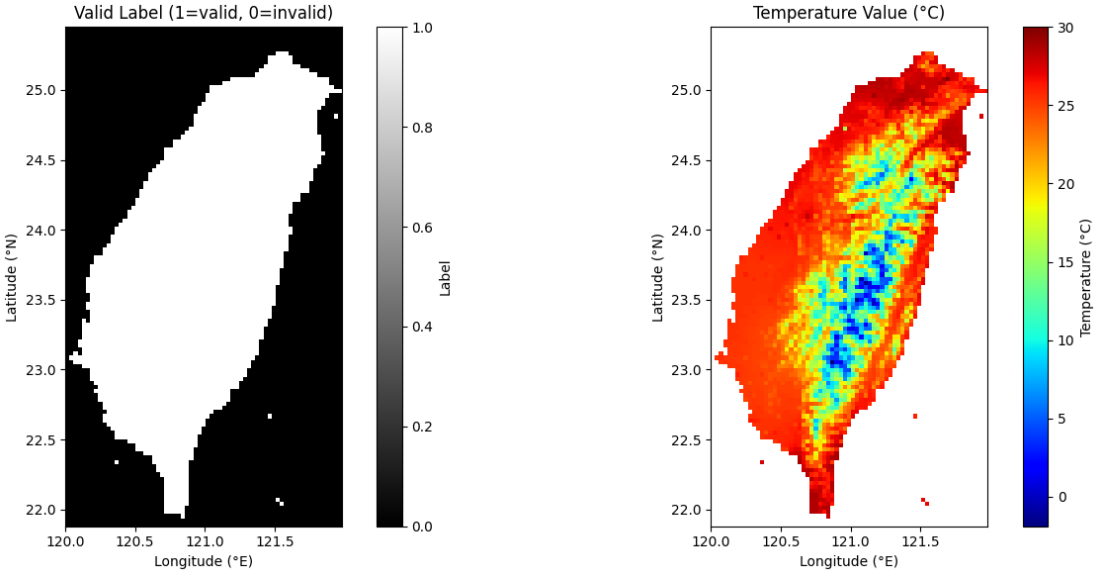
一、資料處理

使用 O-A0038-003.xml 資料集中的資料進行訓練。

將資料分成兩組數據：

Classification 資料集	Regression 資料集
資料型態：(經度,緯度,標籤) 溫度有效：label = 1 溫度無效：label = -1	資料型態：(經度,緯度,溫度) 溫度無效：Nah

輸入結果：



由於經緯度數值範圍相對較大，若直接輸入容易影響效率，因此我們將經緯度資料正規化 (Normalization) 至 [0,1]，以提升模型的收斂速度與數值穩定性。同時設定 80% 資料為 training data，20% 資料為 validation data。

二、模型設計

Classification	Regression
輸入：(經度、緯度) 輸出：1 或 0 結構：6 層(2-128-64-32-8-1) 激活函數：ReLU()	輸入：(經度、緯度) 輸出：溫度 結構：5 層(2-64-32-8-1) 激活函數：Tanh()

三、訓練過程

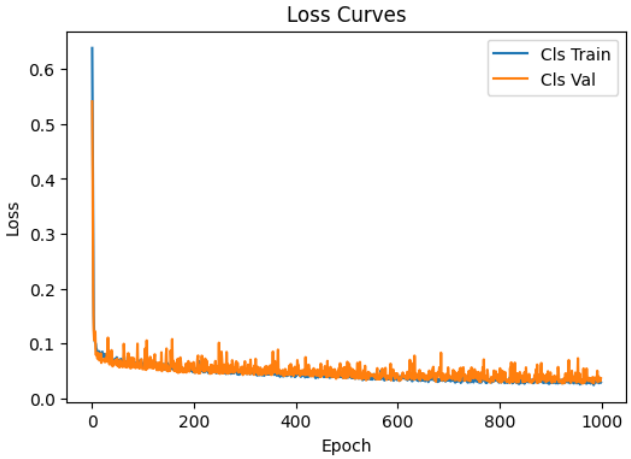
Batch = 64

Classification	Regression
Optimizer：Adam lr=1e-3 Loss function: BCE loss Epochs：1000	Optimizer：Adam lr=1e-3 Loss function: BCE loss Epochs：1000

四、結果分析

(一) Classification

Loss:




```
[Epoch 100] Cls Train=0.0575 | Cls Val=0.0533
[Epoch 200] Cls Train=0.0563 | Cls Val=0.0661
[Epoch 300] Cls Train=0.0462 | Cls Val=0.0611
[Epoch 400] Cls Train=0.0394 | Cls Val=0.0489
[Epoch 500] Cls Train=0.0385 | Cls Val=0.0369
[Epoch 600] Cls Train=0.0346 | Cls Val=0.0454
[Epoch 700] Cls Train=0.0325 | Cls Val=0.0565
[Epoch 800] Cls Train=0.0289 | Cls Val=0.0299
[Epoch 900] Cls Train=0.0281 | Cls Val=0.0387
[Epoch 1000] Cls Train=0.0293 | Cls Val=0.0364
```

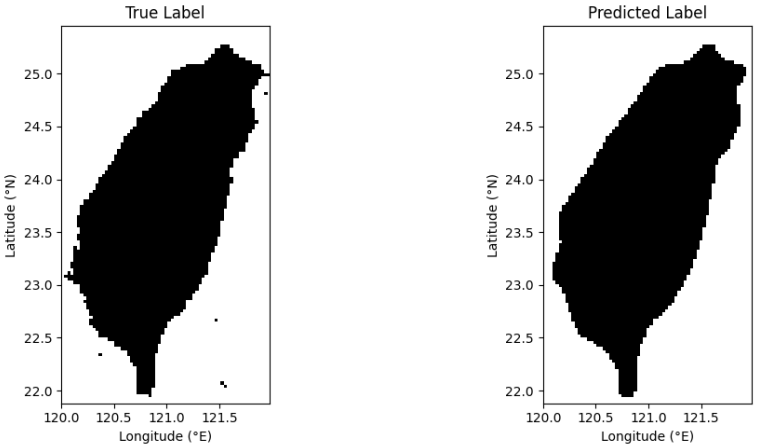
在初期 (Epoch < 50)時，收斂非常快速

而 Epoch>50 時，變動卻沒有太大起伏，只有微小震盪。

並且在學習後期，val loss 也明顯不高。

因此從 loss 可以看出學習的結果算還不錯。

Result:



將預測結果畫成圖片並與實際資料作為比較，發現結果非常不錯。

預測結果能夠畫出一個像台灣土地的輪廓，但離島區域卻沒有辦法訓練出來。

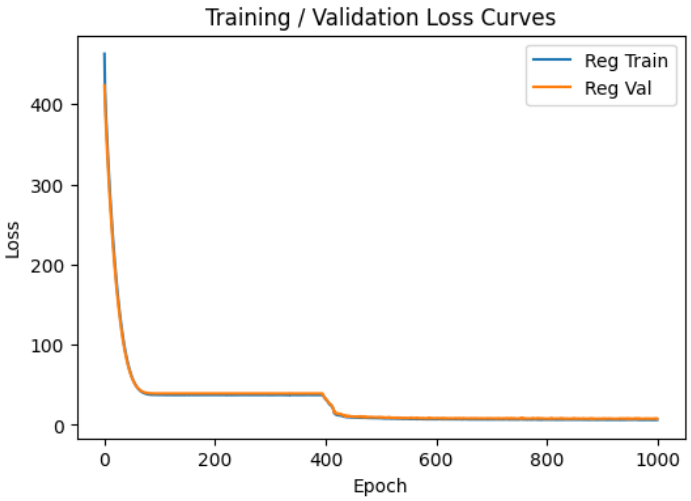
我認為或許調整模型、收斂速度能夠訓練出離島區域的輪廓。

或者將訓練座標網格並得更多(拆分更細)也有機會訓練出更好的結果。

但缺點是訓練時間也會跟著拉長。

(二) Regression

Loss：



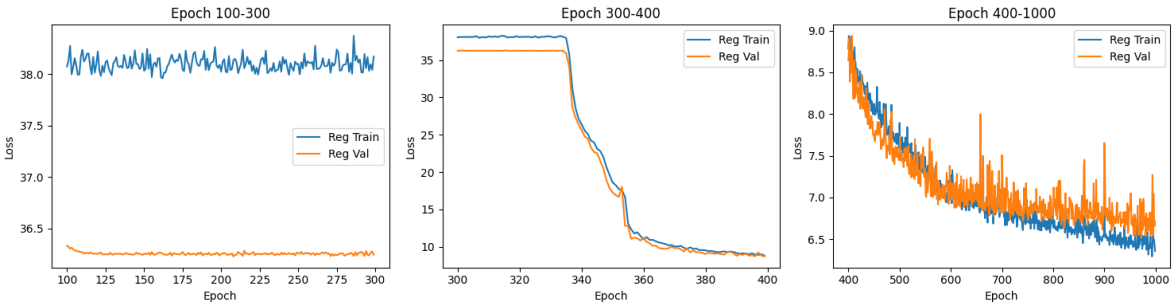

```
[Epoch 100] Reg Train=37.8016 | Reg Val=38.1880
[Epoch 200] Reg Train=37.5484 | Reg Val=38.1745
[Epoch 300] Reg Train=37.7429 | Reg Val=38.1742
[Epoch 400] Reg Train=7.9609 | Reg Val=9.1383
[Epoch 500] Reg Train=7.3498 | Reg Val=8.9008
[Epoch 600] Reg Train=6.8996 | Reg Val=8.0714
[Epoch 700] Reg Train=6.6385 | Reg Val=7.7711
[Epoch 800] Reg Train=6.5099 | Reg Val=7.8035
[Epoch 900] Reg Train=6.3762 | Reg Val=7.6460
[Epoch 1000] Reg Train=6.1991 | Reg Val=7.5901
```

在初期 (Epoch < 50)時，收斂快速。

而 Epoch 在 50-350 時，變動卻沒有太大起伏。

但 Epoch 在 350-400 時，又突然急速下降。

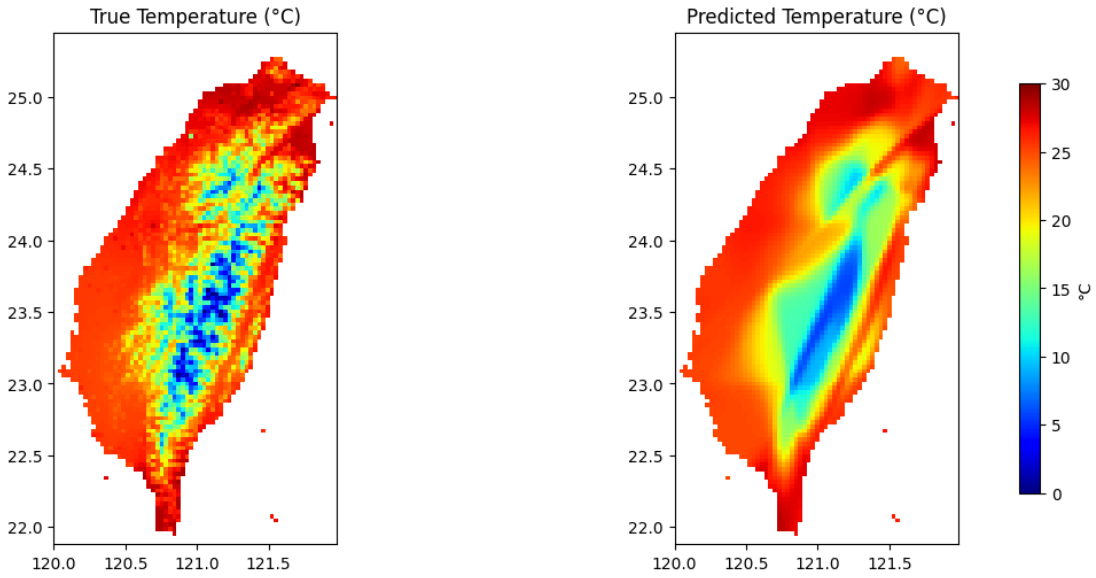
為了觀察這個現象，將 loss 不同段落畫出來。



發現到在 Epoch 340 之前都很平穩，且 loss 約為 38 左右。

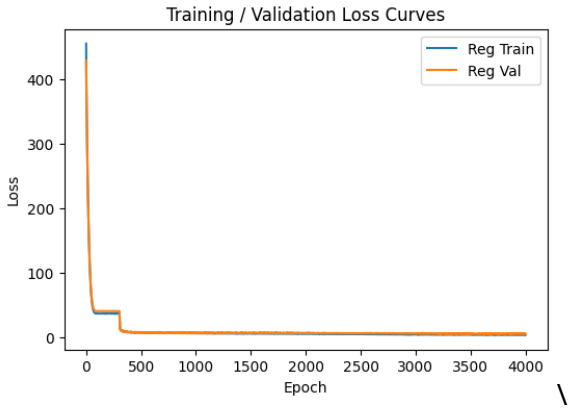
但在 340-360 之間，loss 下降至 8。且 360 之後穩定下降。

Result：



在這個情況下，做出來的結果還不錯，但是由於看到 loss 還在下降，因此想觀察結果會不會更好。因此有了 loss2 跟 result2。使用 Epochs =4000 計算。

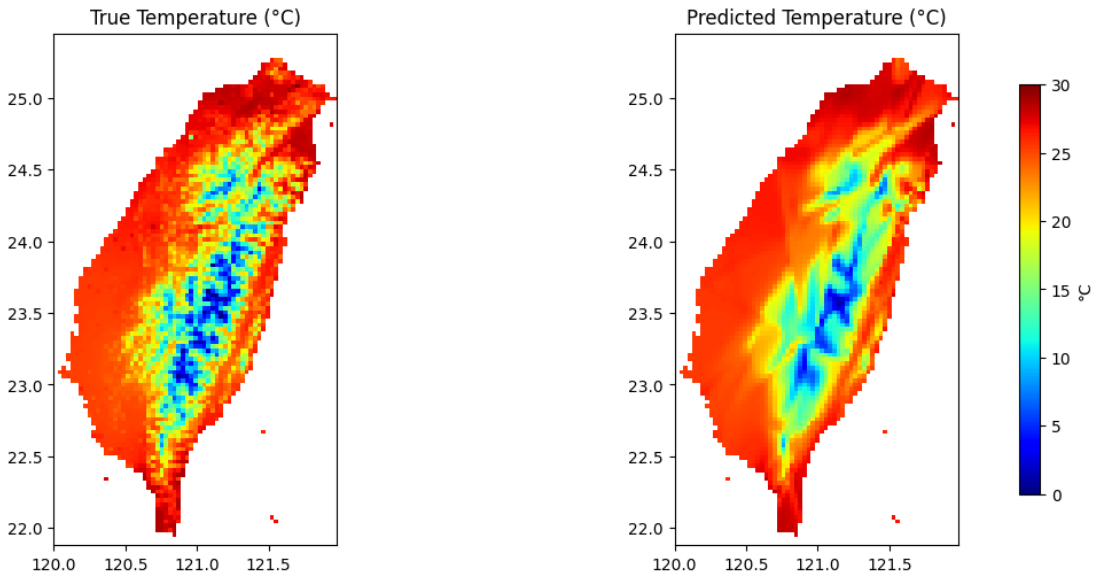
Loss2：



\

Loss 前半段跟 Epoch1000 差不多，後半段持續下降，最後停在 3.7 左右。

Result2：

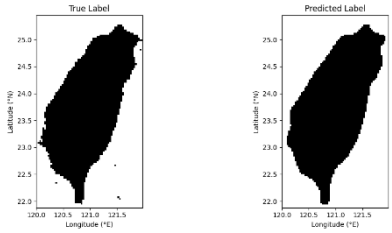


運行 4000 次的預測結果比 1 運行 1000 次精準，細節也更多。

五、結論

(一) Classification

我認為做出來的結果非常不錯，不管從 loss 或 result 都蠻好的，但或許可以改變一些方法，像是網格數變多、訓練次數變多同時與速率更小，讓結果變得更好。



(二) Regression

做出來的結果也很不錯，模擬出的溫度圖很像實際的資料。但 loss 卻有一個特別的結果，有一個像階梯的東西，我認為或許是因為卡在了局部最小值，但隨著模型更新，離開了局部最小值並繼續更新，才導致這樣的結果。