

111652041 蔡翔宇 ML_HW2 程式報告

1. 我們使用 Neural network 的技術逼近一個連續函數： $\frac{1}{1+25x^2}$ 。為什麼這裡不使用傳統的高次插值多項式來逼近呢？用高次多項式插值容易出現 **runge phenomenon** (i.e. 趨於端點時邊界附近震盪很大)，因此我決定在這裡使用 MLP (Multi-Layer Perceptron) 並使用的激活函數為 **tanh** 來逼近平滑連續函數，這樣比較不會像高次多項式那樣在邊界爆震。

2. 而我們亦可以從 Universal Approximation Theorem (萬能近似定理) 可以知道，若我們的單個 hidden layer 中只要神經元 (neuron) 夠多，並且激活函數 (activation function) 足夠好，那麼我們就可以逼近任意的連續函數。(在一個有界的區間，任意小的誤差)：

$f: R^n \rightarrow R$ is a continuous function,

then, exist a single neural network of hidden layer:

$$F(x) = \sum_{i=1}^m \alpha_i \sigma(w_i x + b_i),$$

s.t. $\forall \varepsilon > 0, |f(x) - F(x)| < \varepsilon, \forall x \in K$, where K is the domain.

3. 所以我們使用 NN 的技術來學習連續函數： $\frac{1}{1+25x^2}$ ，而我們的學習方式也就是給許多的樣本點 $(x, f(x))$ ，讓 NN 來輸出 $f^{\wedge}(x)$ ，然後使用 MSE 作為我們的損失函數：

$$L(\theta) = \frac{1}{N} \sum_{i=1}^N (f^{\wedge}(x_i) - f(x_i))^2$$

並且在此使用的是 ADAM 梯度下降更新參數 θ ，直到驗證集不再進步 (Early stopping) 為止。

4. 首先我們先產生樣本點並且切分，而我們的 model 選擇最簡單的均勻採樣，並切成訓練集與測試集來做對照：訓練用來學參數，驗證用來挑模型，(Early stopping) 測試最後才看學習的泛化表現。
而為什麼需要有訓練集與測試集呢？就是樣讓機器早停 (Early stopping)，避免過擬合 (Overfitting)，過擬合也就是完完全全只有紀錄訓練的數據。
5. 再來我們在模型設計的結構選取 $1(\text{input}) \rightarrow 64 \rightarrow 64 \rightarrow 1(\text{output})$ ，而我們在

這裡選的 **activation function** 是 **tanh**，原因是 **tanh** 在 **range (-1,1)**對於平滑函數比較逼近穩健，而程式的輸出層(**output layer**)並不需要活化，因為我們要回歸實數。

6. 接下來就可以計算 **Loss function(MSE)**讓模型朝向平均誤差平方最小的方向去更新。下一步就是訓練流程與早停了:
 - (1) 單一 **epoch** 的流程(i.e. $\theta \leftarrow \theta - \eta \nabla_{\theta} L_{train}(\theta)$, where η is learning rate)
 - (2) 驗證我們的相位(不更新權重)，只做 **forward**，檢查泛化能力。
 - (3) **Early stopping**：當驗證損失持續不進步超過一定值(**patience**:耐心值)時，就停。防止我們的模型變得越訓練越把訓練集背起來而傷害泛化。結束後，把最佳的權重載回模型。
 - (4) 測試與評估：不要只看訓練和驗證，最後要用「測試集」檢查模型泛化能力因為驗證集已經在訓練過程中影響了模型(**By Early stopping**)，所以最後必須用完全沒參與過模型選擇的測試集來計算最終的表現指標。
 - (5) 如果 **MSE**：看整體平均表現與 **Max error**：看最差情況表現都小，代表我們做出來的模型非常穩定。

