

(一) 目的：

甲、用神經網路學習並預測 $f(x) = 1 / (1 + 25x^2)$ 在 -1 到 1 之間的函數值

(二) 資料：

甲、用 numpy 函式庫中的均勻隨機取點

乙、將資料數量的六成作為訓練資料，三成作為驗證資料，一成作為測試資料

(三) 結構：

甲、使用四層的神經網路，兩個隱藏層的神經元都是十個，輸入 x 值後預測並輸出 y 值（輸入和輸出都是數字而非向量）

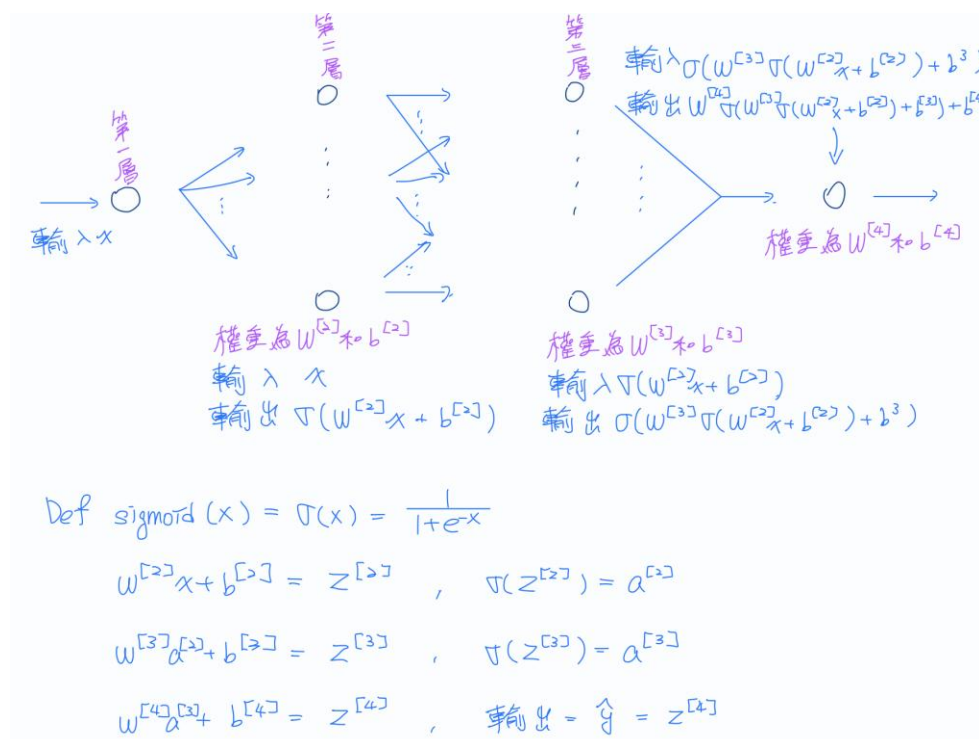
乙、啟動函數使用 $\text{sigmoid}(x) = 1 / (1 + e^{-x})$

(四) 方法：

甲、設定目標函數 $f(x) = 1 / (1 + 25x^2)$ ，並在 -1 到 1 之間均勻隨機取 2000 個點，並將六成作為訓練資料

乙、設定神經網路（四層）和啟動函數 $\text{sigmoid}(x) = 1 / (1 + e^{-x})$

丙、設定初始權重（在 -5 到 5 之間隨機取值）



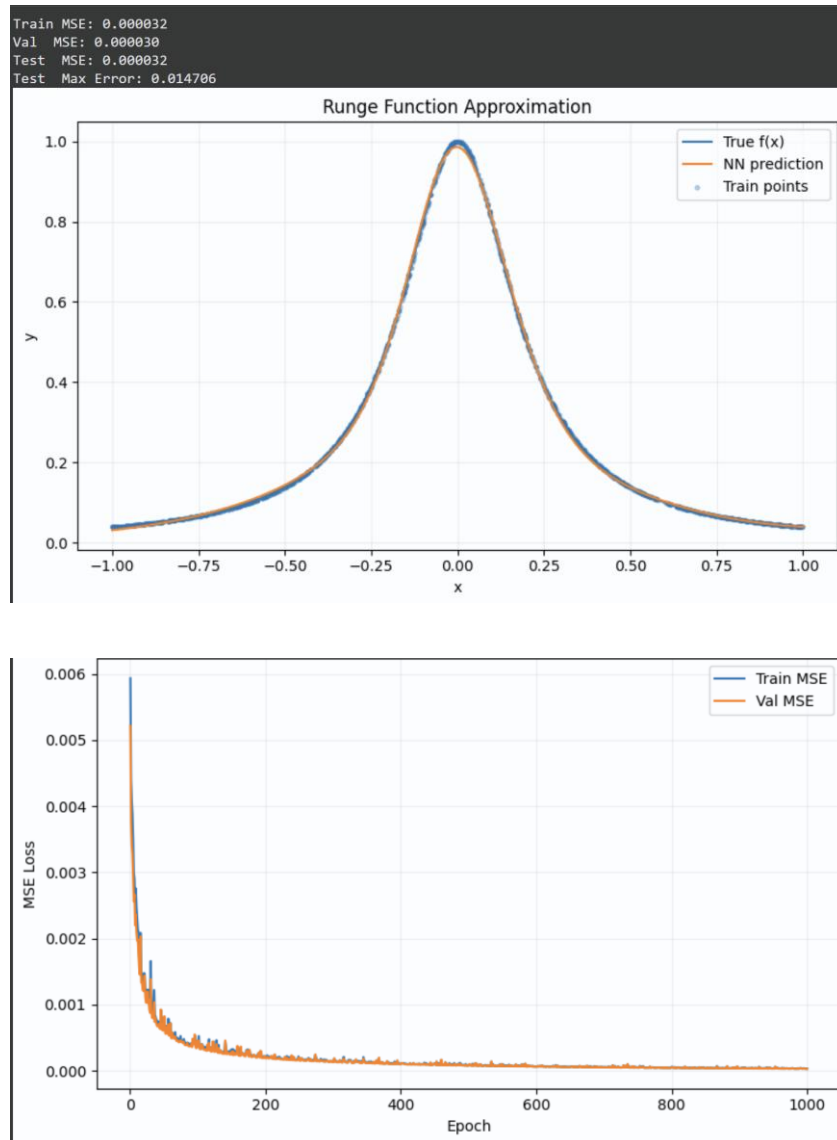
丁、開始訓練神經網路（放入訓練資料）

戊、利用反向傳播計算出梯度 ∇loss （ loss 為損失函數）

己、讓權重移動 $lr * -\nabla \text{loss}$ (lr 為學習率，這裡設定為 0.05)

庚、每次使用四筆資料來學習，學習完 2000 筆資料稱為一輪，總共學習 1000 輪

(五) 結果：



(六) 討論：

甲、上周五去找老師您報告的時候我的程式寫錯了，我誤將輸出的預測值放進了 $\text{sigmoid}(x)$ ，導致只用很少的資料就有還能接受的精度 (60 筆訓練資料，每批次放入 5 筆，做 70 輪，學習率為 0.5，最後最大誤差為 0.1)，導致此現象的原因還在思考。