機器學習作業報告_111652026劉馥瑞

1. 生成資料
    a. 定義Runge function。
    def runge(x):
        return 1.0 / (1.0 + 25.0 * x**2)
    b. 在 [−1,1]隨機抽取樣本點，總共 1400 個，y 值就是 Runge function。
    N_train, N_val, N_test = 1000, 200, 200
    xs = np.random.uniform(-1, 1, size=(N_train + N_val + N_test, 1))
    ys = runge(xs)
    c. 打亂資料，避免有序性影響訓練，再切分成訓練集1000 筆、驗證集200 筆、測試集200 筆。
    perm = np.random.permutation(len(xs))
    xs, ys = xs[perm], ys[perm]
    d. 接著轉成 PyTorch Tensor 並放到 GPU。
2. 模型
    a. 設計一個多層感知機，結構是1→50→50→1，並使用tanh為激活函數。(輸入一個 x，會輸出對應的近似值 f(x)。)
    class MLP(nn.Module):
        def __init__(self, hidden_sizes=[50,50], activation=nn.Tanh):
            super().__init__()
            layers = []
            in_dim = 1
            for h in hidden_sizes:
                layers.append(nn.Linear(in_dim, h))
                layers.append(activation())
                in_dim = h
            layers.append(nn.Linear(in_dim, 1))
            self.net = nn.Sequential(*layers)
        def forward(self, x):
            return self.net(x)
3. 訓練設定
    a. 損失函數為均方誤差，最佳化器為Adam(學習率1e-3)，最多跑2000epoch，並且設定早停 (如果驗證集連續 100 個 epoch 沒改善，就停止)。
    訓練過程：
    逐批訓練 (batch size=64)，每個 epoch 都計算 train loss 與 val loss，並保存最佳驗證集模型。
    loss_fn = nn.MSELoss()
    optimizer = torch.optim.Adam(model.parameters(), lr=1e-3)
    n_epochs = 2000
    patience = 100
4. 測試與評估
    a. 用最佳模型在測試集上計算測試 MSE與最大誤差。
    with torch.no_grad():
        pred_test = model(x_test_t).cpu().numpy().ravel()
        mse_test = np.mean((pred_test - y_test.ravel())**2)
        max_err = np.max(np.abs(pred_test - y_test.ravel()))

5. 輸出Runge function、神經網路近似曲線、測試點的預測結果，再畫出訓練/驗證損失曲線，觀察收斂情況。

輸出結果&分析(生成的兩張圖放在Week_2的資料夾中)

```
Epoch 1: train_loss=0.074912, val_loss=0.070516
Epoch 200: train_loss=0.000003, val_loss=0.000003
Early stopping at epoch 399
Test MSE: 6.655059e-07, Test max abs error: 2.355876e-03
```

1. Epoch1
   一開始訓練誤差 (train_loss=0.0749) 與驗證誤差 (val_loss=0.0705) 都不小，代表模型初始還沒學好 Runge function。
2. Epoch200
   訓練與驗證誤差已經下降到 3e-06 (非常小的誤差)，說明 NN 幾乎完美擬合了 Runge function。
3. Early stopping at epoch 399
   早停在第 399 個 epoch 就觸發，代表模型的驗證集誤差已經不再顯著改善，訓練可以停止。這避免了 overfitting。
4. Test MSE ≈ 6.7×10^{-7}
   測試均方誤差非常小，學到的函數近似真實 Runge function。
5. 最大絕對誤差 ≈ 0.00236
   在所有測試點中，最差的一點預測誤差大約是 0.00236，而 Runge function 的函數值範圍在 [0, 1] 之間。相對誤差 ≈ 0.236%。(非常小)

Loss曲線分析

1. 趨勢分析
   a. 訓練損失(Train Loss)
      隨著訓練週期的增加，訓練損失持續下降。代表模型有效地從訓練資料中學習，從而降低預測誤差。剛開始損失值會快速下降，然後逐漸趨於平穩，這是一個健康且預期中的趨勢。
   b. 驗證損失 (Validation Loss)
      驗證損失在訓練初期也會隨週期數增加而下降。但在後續階段，它可能開始波動或保持平穩，甚至輕微上升。這種現象表示模型的泛化能力可能遇到了瓶頸，或已經開始對訓練資料過度擬合。
2. 關鍵觀察
   a. 收斂性
      訓練損失曲線呈現平滑下降，顯示模型在訓練資料上收斂良好。然而，驗證損失並未持續下降(尤其在訓練後期)，這暗示模型的泛化能力有限。
   b. 過度擬合跡象
      訓練後期模型已過度學習訓練資料的細節，導致其在未見過的資料上表現不佳。
   c. 對數尺度
      均方誤差使用對數坐標，這表示損失值的變化範圍很大，顯示模型在訓練初期學習得非常迅速，使損失值得以快速下降。

注:本次作業的code跟資料分析有使用chatgpt作為輔助工具。