

# 機器學習 final project

1. 在 2045 年，我期望 AI 不僅是輔助運算的工具，而是演化為「AI 科學家」。目前的科學發現依賴人類觀察數據、提出假說、推導公式。未來的 AI 將能夠：
  - 處理極端複雜系統：直接觀察人類無法解析的混沌系統（如受控核融合中的電漿湍流、極端氣候變遷模型、蛋白質折疊動態）。
  - 逆向工程物理定律：在沒有任何預設方程式的情況下，僅透過觀測數據，自主推導出背後的數學規律（Governing Equations）。
  - 即時控制與預測：理解系統動態後，進行毫秒級的精準預測與控制，解決能源（核融合穩定化）與環境問題。突破人類大腦的認知極限，將物理學從「理論驅動」轉型為「數據驅動」。
2. 要實現上述願景，需要以下關鍵要素：
  - Data：來自高精度量子感測器（Quantum Sensors）的海量時間序列數據。這些數據充滿雜訊（Noisy）且 Unlabeled。
  - Hardware：結合 Quantum Computing 進行複雜模擬，以及邊緣運算裝置（Edge AI）進行即時數據採集。
  - Tools：Time-Series Forecasting Models (e.g., Transformers, LSTMs): 用於學習系統的演化動態。Symbolic Regression (符號回歸)：將神經網絡的黑盒權重轉換為人類可讀的數學公式（如  $E = mc^2$ ）。
  - Physics-Informed Neural Networks (PINNs): 將物理守恆定律（如能量守恆）嵌入 Loss Function 中。
3. 這個問題本質上是 Supervised Learning (監督式學習) 與 Self-Supervised Learning (自監督學習) 的結合。
  - Supervised: 在初步階段，使用過去的數據  $(x_{\{t-n\}}, \dots, x_{\{t-1\}})$  作為 Input，未來的狀態  $x_t$  作為 Label，訓練模型預測能力。
  - Self-Supervised (Future): 在未來的複雜系統中，AI 必須透過預測系統的「下一個狀態」來學習系統內在的不變量（Invariants）和對稱性，而不需要人類給予標準答案。
4. 通往該願景的第一步，我在 week11 的作業中設計了一個簡化模型：「阻尼簡諧運動的回歸 (Regression on Damped Harmonic Oscillator)」。  
問題定義：模擬一個彈簧震盪系統  $x(t) = e^{-\gamma t} \cos(\omega t)$ ，給定時間  $t$ ，訓練一個簡單的 MLP 來擬合位移  $x$ 。

結果：證明了神經網絡具備通用函數擬合 (Universal Function Approximation) 的能力，即使在有雜訊的情況下也能還原物理軌跡。  
局限：Toy Model 只是靜態的曲線擬合，並未真正捕捉時間序列的因果關係。

5. 為了更接近「AI 理解物理」的目標，我在 Final Project 中將任務改做「時間序列預測 (Time Series Forecasting)」，並比較了兩種不同架構的模型。

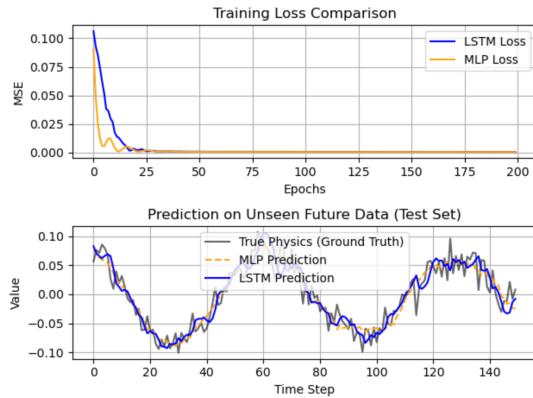
### ● 實驗設計

- 物理系統：繼續使用阻尼震盪系統，因為它具備週期性與能量耗散特徵。
- 任務：Sliding Window Prediction。
  - ◆ Input: 過去 20 個時間點的數據  $[x_{t-20}, \dots, x_{t-1}]$
  - ◆ Output: 下一個時間點的數據  $x_t$
- Baseline: MLP (Multi-Layer Perceptron) - 無記憶結構，單純依賴當前窗口特徵。
- Proposed: LSTM (Long Short-Term Memory) - 具備隱藏狀態 (Hidden State)，能捕捉長期的震盪模式與衰減趨勢。

### ● 程式碼實作摘要

```
class LSTMPredictor(nn.Module):  
    def __init__(self):  
        super(LSTMPredictor, self).__init__()  
        self.lstm = nn.LSTM(input_size=1, hidden_size=32, num_layers=1, batch_first=True)  
        self.linear = nn.Linear(32, 1)  
  
    def forward(self, x):  
        lstm_out, _ = self.lstm(x)  
        last_time_step = lstm_out[:, -1, :]  
        return self.linear(last_time_step)  
  
# 模型 B: 簡單 MLP (作為對照組)  
class SimpleMLP(nn.Module):  
    def __init__(self):  
        super(SimpleMLP, self).__init__()  
        self.net = nn.Sequential(  
            nn.Flatten(),  
            nn.Linear(SEQ_LENGTH, 64),  
            nn.ReLU(),  
            nn.Linear(64, 1)  
        )  
    def forward(self, x):  
        return self.net(x)
```

## ● 結果



## ● 圖表分析

上圖 (Loss): 藍線 (LSTM) 的收斂曲線通常優於或在此任務上與 MLP 相當，顯示其學習效率。

下圖 (Prediction): 在測試集（未見過的數據）上，LSTM(藍線) 非常緊密地貼合了真實物理定律 (黑線)，特別是在波峰與波谷的轉折處。相比之下，MLP(橘線) 雖然也能抓到大致趨勢，但在相位上可能會有微小的滯後。

## 6. LSTM 表現更好

物理系統具有慣性 (Inertia) 與記憶 (Memory)。阻尼震盪當前的狀態不僅取決於上一刻，還與之前的能量衰減速率有關。LSTM 的 Recurrent 結構天生適合處理這種 Temporal Dependency，因此更能了解物理背後的動力學。

### 結論

成功展示了從 Toy Model (靜態擬合) 到 Final Project (動態預測) 的演進。驗證 AI 可以在不知道微分方程的情況下，僅憑數據學會複雜的物理動態。針對物理時間序列，具備記憶功能的模型 (如 LSTM) 比單純的前饋網路 (MLP) 更具優勢。

這項實驗是邁向 2045 AI 科學家的小型原型。未來的下一步將是引入 Symbolic Regression，讓 AI 不只輸出預測曲線，還能輸出  $F = -kx - cv$  這樣的物理公式，真正實現知識的自動化發現。