Written assignment

1. Week 1 Question

Q.梯度下降可能卡在 local minima 非 Global minama，在高維度時 saddle point 和 plateau 導致學習緩慢

A. 現代研究指出，在高維空間（High-dimensional space，即參數非常多的深度神經網路）中，問題通常不是 Local Minima，而是 鞍點（Saddle Points）。而 Saddle point 問題則可以使用帶有 動量（Momentum） 的優化器（如 SGD with Momentum 或 Adam），利用過去的慣性衝過平坦的鞍點區域。

相關文獻：Dauphin, Y. N., et al. (2014). "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization". NeurIPS.

Q.如何找到適合的 lr?

A. 在訓練開始前，先跑一個極短的 epoch。讓 LR 從非常小（如 1e-7）指數增長到非常大（如 10）

相關文獻：Smith, L. N. (2017). "Cyclical Learning Rates for Training Neural Networks". IEEE WACV.

2. Week 2 Question

Q. 對於二元分類問題，loss function 好像更常使用 Cross-Entropy，MSE 在（one-hot encoder)下 假如使用 sigmoid：$\sigma(z)$的導數是$\sigma(z)(1-\sigma(z))$ 真實為 1 預測輸出非常錯誤 假設為 0.01 時 就會導致梯度變得非常小，幾乎為零 因為 MSE 假設預測目標是連續的，且誤差服從高斯分佈。但分類問題是離散的，One-hot encoder 雖然轉化為數值向量，但不知道理論上這樣是不是合理的？

A. 雖然 One-hot encoding 把離散變量變成了向量，但它仍然代表機率分佈，而非連續數值。因此，使用 MSE 在理論上是「模型假設與數據本質不符」，在實作上則會導致學習停滯。

相關文獻：Golik, P., et al. (2013). "Cross-entropy vs. squared error training: a theoretical and experimental comparison". Interspeech.

Q. Approximation Theory 提到可以用 p 個神經元來近似 x 2 p − 1 ，如

果要近似更高次的多項式，是否需要更多神經元的網路，和網路深度是否效果不同？

A.取決於要增加深度還寬度，如果是深度：若第一層做X^2

，第二層做$((X^2)^2 = x^4$第 p 層就能達到X^{2p}。只需 O(p)個神經元

如果是淺層，根據 Universal Approximation Theorem，一個只有單隱藏層的淺層網路確實可以近似任何函數。 但是為了近似像 x^{2p-1}

這樣的高階多項式或高頻震盪函數，淺層網路需要 指數級（ 2^p ）的神經元數量。

3. Week 3 Question

Q. 在網絡架構中使用奇函數或週期函數（如 sin ,cos sin,cos）是否有幫助？

A. 例如 Tanh 就是奇函數 ($f(-x) = -f(x)$。它的優點是 Zero-centered (以零為中心)。這意味著輸出的期望值接近 0，這對於下一層的梯度傳播非常有利，比起 Sigmoid (輸出恆正) 能收斂得更快。

相關文獻：Sitzmann, V., et al. (2020). "Implicit Neural Representations with Periodic Activation Functions" (SIREN). NeurIPS

Q. 對於 p 很大的X^P，神經元數量和網路深度是否會有訓練困難？

A. 是，會導致極大的訓練困難。

4. Week 4 Question

Q. 為何 Logistic Regression (LR) 不使用最小平方法 (MSE)？如果使用會怎樣？

A. 雖然在數學上可以硬把 MSE 用在 Logistic Regression 上，但在實務上幾乎沒人這樣做，主要原因有兩點：非凸性 (Non-Convexity) 與 梯度消失 (Vanishing Gradient)。

Q. Newton's Method vs. Gradient Descent：何時使用哪一種？

A. 在實務上（如 Scikit-Learn 的 Logistic Regression)，會用 L-BFGS。 這是一種 「擬牛頓法」(Quasi-Newton)。它不真正計算 Hessian 矩陣，而是通過歷次的梯度變化來「估計」Hessian 矩陣。這結合了 Gradient Descent 的低成本與 Newton's Method 的快收斂，是中型數據集的首選。

5. Week 5 Question

Q. GDA 是否比 Logistic Model 更強大或更嚴格？

A. GDA 確實比較「嚴格」(Stricter)，依賴更強的假設

Q. 既然更嚴格，訓練效果一定會更好嗎

A. 不一定。這取決於數據是否真的符合 GDA 的假設

相關文獻：Ng, A. Y., & Jordan, M. I. (2001). "On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes". NIPS.

6. Week 6 Question

Q. 在偶函數的例子中，即使神經網路 N 使用了無限可微的激活函數 tanh，建構出的 u(x)=N(|X|)，但在 x=0 處不可微，這樣 feedforword fully-connected neural network 的 regularity 還會和其他激活函數相同嗎?

A. 不會。整個模型 u(x)的正則性會被破壞，它不再是 C∞（無限可微)，在 x=0 處會降級為C^0（僅連續但不可微)。

7. Week 7 Question

Q. 在 DSM 中，加入的雜訊強度 σ 是否會影響模型的學習效果？如果雜訊太小或太大，會有啥效果?

A. σ小→只有地圖上的道路有路標，野外沒有。一旦迷路（隨機初始化）就回不來了。σ大→整個地球都被標記成「往地心走」。你能找到地球，但找不到紐約市（細節丟失)。多尺度σ(Diffusion)→先用衛星導航到城市，再用街道圖導航到門牌。

8. Week 8 Question

Q. 在 SSM 中，會將 SCORE 投影到一個隨機向量 V 上，p(V)除了球面均勻分布，能否使用其他分布?

A. 可以，而且選擇不同的分佈確實會影響訓練的「穩定性」，但不會改變「理論上的最佳解」。

9. Week 10 Question

Q. 在推倒 SDE 對應 ODE 時，如何確保解的唯一性和穩定性?

A. 要確保 SDE 和 ODE 描述的是同一個機率分佈演化過程需要透過 Fokker-Planck 方程的等價性和 Trajectory Uniqueness，穩定性則靠神經網路的平滑設計 以及 先進的數值求解器 (DPM-Solver) 來克服剛性問題

相關文獻：Song, Y., et al. (2021). "Score-Based Generative Modeling through Stochastic Differential Equations". ICLR.（這篇論文的 Appendix D）

Gyöngy, I. (1986). "Mimicking the one-dimensional marginal distributions of processes having an Itô differential".

Probability Theory and Related Fields.

- **Toy model/Solvable Model Problem in your <u>final project</u>**

First Step of an AI Physicist: Rediscovering Damped Dynamics from Data

1. 前言：20 年後的願景

在我的 Final Project 構想中，預測 20 年後的 AI 將演化為「The Autonomous Scientific Discoverer」。屆時的 AI 不再僅僅是執行人類指令的計算工具，而是能夠直接觀察極端複雜的自然現象（例如：受控核融合中的電漿湍流、極端氣候的混沌系統），並在沒有人類預設方程式的情況下，自主推導出背後的物理定律與數學模型。

為了實現這個願景，必須從最基礎的物理單元開始驗證。如果 AI 無法理解簡單的機械震盪，它就不可能理解複雜的量子力學。因此，先設計一個可行的「簡化模型問題」，作為通往該能力的第一步。

2. 設計思路

為了模擬 AI「觀察現象並發現規律」的過程，我選擇了經典物理學中的「阻尼簡諧運動 Damped Harmonic Oscillator」作為本次的簡化模型。

為什麼選擇這個模型？

單純的線性回歸過於簡單，無法體現物理世界的複雜性。而阻尼震盪系統包含了兩個動力學系統的核心特徵：

週期性 (Periodicity)： 代表系統的震盪與波動特性。

能量耗散 (Dissipation)： 振幅隨時間呈指數衰減，代表摩擦力或阻力的存在。

其背後的物理方程為二階微分方程：$md^2(x)/dt^2 + cdx/dt + kx = 0$
。目標是：在不告訴 AI 這個微分方程的前提下，給它看一堆帶有雜訊的觀測數據，看它能否自己「學」會這個運動軌跡。

實作方法 (Methodology)

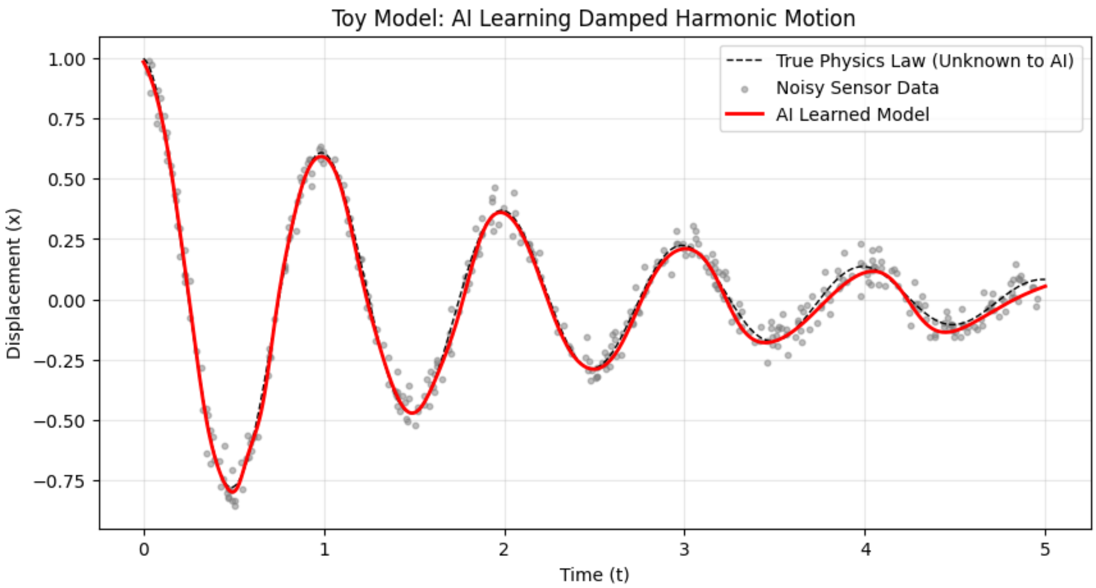
3.實作方法

我使用 Python 與 PyTorch 建構了一個深度神經網絡 (DNN) 來執行此任務。

數據生成 (Data Generation)：

我模擬了一個理想的阻尼震盪函數 x(t)=$e^{-0.5t}cos(2\pi t)$
，並在生成的數據中加入了高斯分佈的隨機雜訊 (Gaussian Noise)，以模擬真實世界感測器的誤差。這確保了 AI 必須學習「訊號本身的結構」

而非死背數據點。
模型架構 (Model Architecture)：
我設計了一個全連接神經網絡 (MLP)。為了捕捉複雜的波形特徵，我選擇了 Tanh (雙曲正切) 作為激活函數，而非一般的 ReLU。因為 Tanh 的輸出範圍在 -1 到 1 之間，且具有平滑的非線性特徵，非常適合處理物理波動訊號。
訓練策略 (Training)：
輸入為時間 t，輸出為位移 x
。使用均方誤差 (MSE Loss) 作為損失函數，透過 Adam 優化器進行 3000 次迭代訓練。
4. 結果



上圖展示了模型的訓練結果。灰色散點代表帶有雜訊的觀測數據，黑色虛線代表真實的物理定律 (AI 未知)，而紅色實線則是 AI 的預測模型。
從圖中可以觀察到幾個關鍵點：
抗噪能力： 儘管輸入數據充滿雜訊，AI 依然成功學會了平滑的軌跡，沒有發生過度擬合 (Overfitting) 去追逐每一個雜訊點。
特徵捕捉： AI 精準地抓住了「震盪頻率」以及「指數衰減的包絡線 (Envelope)」。這意味著神經網絡內部已經隱含地構建了類似阻尼震盪的數學邏輯。
5. 結論
這個 Model 成功驗證了「數據驅動物理建模」的可行性。它證明了即使是簡單的神經網絡，也具備通用函數擬合 (Universal Function Approximation) 的能力來逼近微分方程的解。
這雖然只是一個簡單的彈簧系統，但它確立了 Final Project 的核心邏輯：「觀測數據

→神經網絡 →物理規律」。在接下來的 Final Project 中，我將進一步探討如何將此概念擴展到多變數系統，並引入符號回歸 (Symbolic Regression)，讓 AI 不僅能畫出曲線，還能直接寫出 F=ma 這樣的公式，真正實現「AI 科學家」的願景。