本文針對課程 Week 1 至 Week 10 累積之未解問題進行文獻查證。經確認，這些問題在現有機器學習與優化理論中多已有成熟解答。以下列出理論解析與對應之關鍵參考文獻。

Week 1: 基礎優化 (Optimization Basics)

Q1 & Q2: 學習率選擇與自動調整

解答：早期的研究依賴手動調整 (Learning Rate Schedules)，但現代深度學習主要使用「自適應學習率算法 (Adaptive Learning Rate Methods)」。這些方法會根據梯度的歷史統計數據（如一階矩和二階矩）自動調整每個參數的學習率。

- **Reference (Adam Optimizer):**
  Kingma, D. P., & Ba, J. (2014). *Adam: A Method for Stochastic Optimization.* ICLR 2015.
  連結

Q3: 梯度下降陷入局部最小值 (Local Minimum)

解答：現代高維非凸優化 (Non-convex Optimization) 研究指出，深度神經網路的主要困難並非「局部最小值」，而是「鞍點 (Saddle Points)」。鞍點在某些維度是極小值，其他維度是極大值。梯度下降通常能透過隨機性 (SGD) 或動量 (Momentum) 逃離鞍點。

- **Reference:**
  Dauphin, Y. N., et al. (2014). *Identifying and attacking the saddle point problem in high-dimensional non-convex optimization.* NIPS.
  連結

Week 2: 損失函數的選擇 (Loss Functions)

Q: 分類問題為何首選 Cross-Entropy 而非 MSE？

解答：若使用 MSE 搭配 Sigmoid/Softmax 輸出層，當預測值接近 0 或 1 時，導數會趨近於 0，導致「梯度消失 (Vanishing Gradient)」，使訓練極慢。Cross-Entropy ($L_{CE} = -\sum y \log \hat{y}$) 的對數特性正好能抵銷 Sigmoid 的指數特性，保證梯度在誤差大時保持較大數值，加速收斂。

- **Reference:**
  Golik, P., et al. (2013). *Cross-entropy vs. squared error training: a theoretical and experimental comparison.* Interspeech.
  連結

Week 3: 神經網路的構造性逼近 (Constructive Approximation)

Q: 遞迴構造多項式的誤差累積與網路複雜度？

解答：這是神經網路表達能力 (Expressivity) 的核心問題。

1. 誤差累積：雖然遞迴構造會累積誤差，但 Yarotsky (2017) 證明，深度 ReLU 網路逼近多項式所需的參數數量僅隨誤差 ϵ 的對數增長 $O(\text{polylog}(1/\epsilon))$。這顯示深度結構能有效抑制誤差。

2. 複雜度：相比於淺層網路，深層網路在逼近高頻震盪函數或高次多項式時，能指數級地減少所需神經元數量。

- **Reference:**
  Yarotsky, D. (2017). *Error bounds for approximations with deep ReLU networks.* Neural Networks.
  連結

Week 4: 牛頓法與梯度下降 (Newton's Method vs GD)

Q: 何時使用牛頓法優於梯度下降？

解答：牛頓法擁有二次收斂速度 (Quadratic Convergence)，遠快於梯度下降的一次收斂。然而，其計算成本主要在於 Hessian 矩陣的反矩陣 H^{-1}，複雜度為 $O(d^3)$ (d 為參數數量)。

- 結論：當參數量 d 很小 (如 $d < 1000$) 且需要高精度解時，牛頓法較佳。但在深度學習 (參數極多) 中，實務上幾乎全用一階方法 (SGD/Adam)。

- **Reference:**
  Bottou, L., et al. (2018). *Optimization Methods for Large-Scale Machine Learning.* SIAM Review.
  連結

Week 5: 生成模型與判別模型 (GDA vs Logistic Regression)

Q: GDA 與 Logistic Regression 的選擇？

解答：根據 Ng & Jordan (2002) 的經典分析：

1. **GDA (生成模型)**：當數據量較少 (n 小) 時，GDA 收斂速度較快 ($O(\log n)$)，表現通常優於 LR。

2. **Logistic Regression (判別模型)**：當數據量充足時，LR 的漸進誤差較低，且對模型假設錯誤 (數據不完全服從高斯分佈) 更具魯棒性。

- **Reference:**
  Ng, A. Y., & Jordan, M. I. (2002). *On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes.* NIPS.
  連結

Week 6: 多類別 GDA (Multiclass GDA)

Q: 多類別 GDA 該如何實作？

解答：對於 K 個類別，標準作法如下：

1. 計算每個類別的先驗機率 $\phi_k = P(y = k)$。
2. 計算每個類別的平均向量 μ_k。
3. 計算共變異數矩陣 Σ (若為 LDA 則共用，QDA 則計算各自的 Σ_k)。
4. 預測時計算後驗機率 $P(y = k|x)$ 並取最大值。

- **Reference:**
  Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. (Section 4.2).

Week 7: Score Matching 的散度計算

Q: 散度項 $\nabla_x \cdot S(x;\theta)$ 如何在高維度有效計算？

解答：原始 ISM (Hyvärinen, 2005) 計算 Jacobian trace 需要 $O(d^2)$ 計算量，不適用於深度學習。

- 解決方案 **(Sliced Score Matching, SSM)**： Song et al. (2019) 提出利用 Hutchinson trace estimator，將問題轉化為隨機投影，將計算量降至 $O(d)$，使其可應用於高維深度模型。

- **Reference:**
  Song, Y., et al. (2019). *Sliced Score Matching: A Scalable Approach to Density and Score Estimation*. UAI.
  連結

Week 8: 隨機微分方程數值解 (SDE Numerical Methods)

Q: 如何衡量 Euler-Maruyama 方法的好壞？

解答：數值 SDE 的收斂性衡量標準分為兩類：

1. 強收斂 **(Strong Convergence)**：衡量路徑 (Path-wise) 的逼近誤差 $E[|X_t - \hat{X}_t|]$。Euler-Maruyama 為 0.5 階強收斂。
2. 弱收斂 **(Weak Convergence)**：衡量分佈統計量 (Moments) 的逼近誤差 $|E[f(X_t)] - E[f(\hat{X}_t)]|$。Euler-Maruyama 為 1.0 階弱收斂。

- **Reference:**
  Kloeden, P. E., & Platen, E. (1992). *Numerical Solution of Stochastic Differential Equations*. Springer.

Week 10: 逆向 SDE vs 機率流 ODE (Reverse SDE vs PF-ODE)

Q: SDE 與 ODE 在生成時的根本區別與選擇？

解答：在 Score-based Generative Models 中：

1. **SDE (隨機)**：每一步注入噪聲，能修正之前的累積誤差，生成品質通常較高，對 Score 估計誤差較魯棒。
2. **ODE (確定性 Probability Flow)**：允許使用黑盒 ODE Solver (如 Runge-Kutta) 加速採樣；可計算確切的 Log-likelihood；支援潛在空間插值 (Interpolation)。但生成的多樣性可能略低。

- **Reference:**
  Song, Y., et al. (2021). *Score-Based Generative Modeling through Stochastic Differential Equations.* ICLR 2021.
  連結