大三下學期**B101020045** 張芝岑
資工所碩士課程

數位IC課程 DIGITAL IC DESIGN
開課教授:陳培殷 Chen, PeiYin

2024 數位 IC 設計程式專案總結

程式專案 **1:** 模組化加減法器 **(Modular AdderSubtractor)**
目標:設計一個模組化加減法器，能夠在有限域內進行加法和減法運算。

學習內容:
基本概念:
模組化加減法是在指定的模數範圍內進行加法和減法運算。例如，在模數為 13 的情況下，14 和 11 分別被簡化為 1 和 2。
電路設計:
設計包含兩個 ALU 和一個比較器的電路。第一個 ALU 計算中間結果，通過比較器決定下一步操作，第二個 ALU 調整中間結果到最終輸出。
操作選擇:
根據選擇信號進行加法、減法或保持原值操作。

我學到了如何設計一個能夠在特定模數範圍內運算的加減法電路，瞭解了加法和減法操作在不同模數下的特殊處理方法，也學會了如何使用兩個 ALU 和一個比較器來實現該電路，並確保中間結果在正確範圍內，更明白了如何在 ModelSim 中檢查中間結果和比較結果，以及整個 MAS 電路的最終結果。

程式專案 **2:** 登機和取行李系統 **(Checkin and Pickup)**
目標:設計一個有限狀態機來模擬登機和取行李過程。

學習內容:
基本概念:系統使用 FIFO 和 LIFO 結構來模擬機場的乘客登機和行李處理流程。FIFO 用於乘客登機，LIFO 用於行李處理。
系統設計:設計包含兩個 FIFO 和一個 LIFO 的電路架構。乘客和行李數據按特定順序輸入系統，並在系統內部進行管理和處理。
數據處理:設計系統處理乘客和行李數據的輸入輸出，並通過信號控制數據流動和處理順序。

學習了如何設計一個包含 FIFO 和 LIFO 結構的電路系統，用於模擬機場的乘客登機和行李處理流程。理解了 FIFO 和 LIFO 的基本操作及其在這個系統中的應用場景。學會了如何處理乘客和行李數據的輸入輸出，以及如何在系統內部管理這些數據。瞭解了在 ModelSim 中如何驗證 FIFO 和 LIFO 的操作結果，以及整個系統的正確性。

程式專案 **3:** 矩陣乘法器 **(Matrix Multiplier)**
目標：設計並實現一個矩陣乘法器電路。
學習內容：
基本概念：矩陣乘法是數學、計算機科學、工程和物理等領域中的基本操作。矩陣乘法器電路可以處理多行多列矩陣的輸入輸出，並生成結果矩陣。
系統設計：設計能夠執行矩陣乘法的電路，包括矩陣數據的讀寫操作和有效信號、忙碌信號的控制。
操作流程：系統在重置後等待忙碌信號變為低，然後輸入矩陣數據。輸入完成後，根據輸出信號生成結果矩陣。

我學到了如何設計一個能夠執行矩陣乘法的電路，並處理多行多列矩陣的輸入輸出，理解了如何利用時序圖來控制矩陣數據的讀寫操作，以及如何在設計中處理有效信號和忙碌信號，更學會了如何在 ModelSim 中驗證functional simulation和gatelevel simulation的結果，確保電路設計的正確性和性能。

程式專案 **4:** 最大優先佇列 **(MaxPriority Queue)**
目標：設計一個最大優先佇列電路，能夠讀取數據並根據控制命令執行相關操作。
學習內容：
基本概念：最大優先佇列是一種資料結構，能夠根據優先級順序進行數據的插入和提取操作。系統需要支持建立佇列、提取最大值、增加數值和插入數據等操作。
系統設計：設計一個能夠執行上述操作的電路系統，並根據命令信號進行操作控制。
操作流程：根據不同的命令信號，執行相應的操作，並將結果寫入 RAM 進行存儲和檢查。

我學習了如何設計一個能夠執行建立佇列、提取最大值、增加數值和插入數據等操作的電路系統，理解了不同操作命令的具體實現方式及其對應的指令信號，也學會了如何在系統內部處理數據的讀取和寫入，並通過 RAM 進行結果的存儲和檢查，更瞭解了在 ModelSim 中如何驗證佇列操作的正確性和系統的整體性能。

程式專案 **5:** 高級加密標準 **(Advanced Encryption Standard)**
目標：實現 **AES128** 演算法，設計一個硬體架構來加速該演算法。
學習內容：
基本概念：AES128 是一種廣泛使用的加密標準，應用於各種通信場景中。演算法包括多輪的 SubBytes、ShiftRows、MixColumns 和 AddRoundKey 操作。
系統設計：設計一個能夠執行 AES128 演算法的硬體架構，處理輸入數據的加密過程，生成最終的密文。
操作流程：系統在初始輪鍵添加後，通過多輪加密操作轉換狀態矩陣，最後生成密文並輸出。

我學習了 AES128 演算法的基本操作步驟，包括 SubBytes、ShiftRows、MixColumns 和 AddRoundKey，理解了如何將輸入數據轉換為狀態矩陣，並通過多輪加密操作生成最終密文，也學會了如何設計一個能夠通過functional simulation和gatelevel simulation的硬體電路，並在Quartus中生成合成結果，更瞭解了如何在 ModelSim 中驗證加密電路的正確性和性能指標。

這些程式專案不僅加深了我對數位 IC 設計的理解，也提升了我在設計、實作和驗證電路方面的能力。每個程式專案都涉及不同的技術和知識點，從基本的數學運算到複雜的加密演算法，這些學習內容對我的專業知識和實際應用能力都具有很大的幫助。感謝教授給予學習的機會，使我能夠在這些程式專案中取得進步。

在2024學年的數位IC設計課程中，涵蓋了從基本的數學運算電路設計到複雜的加密演算法實現。以下是我在這些程式專案中所學到的詳細內容。

首先，模組化加減法器(Modular AdderSubtractor)的設計讓我理解了如何在有限域(Galois Field)內進行加法和減法操作。具體來說，這個電路需要在特定模數範圍內進行運算，例如當模數為13時，14和11會分別被簡化為1和2。設計過程中，我學會了使用兩個算術邏輯單元(ALU)和一個比較器來實現該電路。第一個ALU計算中間結果，並通過比較器決定下一步操作，第二個ALU調整中間結果到最終輸出。此外，我還學會了在ModelSim中進行功能模擬，檢查中間結果、比較結果和最終結果的正確性。

接著，設計登機和取行李系統(Checkin and Pickup)的過程中，我學到了如何設計一個包含FIFO(FirstIn, FirstOut)和LIFO(LastIn, FirstOut)結構的電路系統，用於模擬機場的乘客登機和行李處理流程。這個系統需要處理乘客和行李數據的輸入輸出，並在系統內部進行管理和處理。具體來說，FIFO用於乘客登機，LIFO用於行李處理。我學會了如何設計系統處理乘客和行李數據的輸入輸出，並通過信號控制數據流動和處理順序。同時，在ModelSim中，我學會了如何驗證FIFO和LIFO的操作結果，以及整個系統的正確性和效率。

在設計矩陣乘法器(Matrix Multiplier)電路時，我學習了如何設計一個能夠執行矩陣乘法的電路，並處理多行多列矩陣的輸入輸出，生成結果矩陣。矩陣乘法是數學、計算機科學、工程和物理等領域中的基本操作。在設計過程中，我學會了如何利用時序圖來控制矩陣數據的讀寫操作，並處理有效信號和忙碌信號。此外，我還學會了在ModelSim中進行功能模擬和門級模擬，確保電路設計的正確性和性能。

在設計最大優先佇列(MaxPriority Queue)電路時，我學會了如何設計一個能夠執行建立佇列(Build_Queue)、提取最大值(Extract_Max)、增加數值(Increase_Value)和插入數據(Insert_Data)等操作的電路系統。這個系統需要根據命令信號進行操作控制，並將結果寫入RAM進行存儲和檢查。我學會了如何處理不同的操作命令，並確保系統能夠正確執行這些操作。同時，在ModelSim中，我學會了如何驗證佇列操作的正確性，並確保整個系統的性能達到要求。

最後，在實現高級加密標準(Advanced Encryption Standard, AES)演算法的過程中，我學習了如何設計一個硬體架構來加速AES128演算法。AES128是一種廣泛使用的加密標準，應用於各種通信場景中。這個演算法包括多輪的SubBytes、ShiftRows、MixColumns和AddRoundKey操作。在設計過程中，我學會了如何將輸入數據轉換為狀態矩陣，並通過多輪

加密操作生成最終密文。此外，我還學會了如何在ModelSim中進行功能模擬和門級模擬，確保電路設計的正確性和性能指標。並在Quartus中生成合成結果，檢查系統的面積和時序性能。

這些程式專案讓我深入瞭解了數位IC設計的各個方面，從基礎的數學運算到複雜的加密演算法，涵蓋了資料結構、時序控制、硬體加速等多個領域，提升了我的設計能力和實作經驗。感謝教授給予學習機會，使我能夠增進自己。

遺傳基因演譯法GENETIC ALGORITHMS
開課教授:郭淑美 Guo, ShuMei

在這段學習過程中，我深入學習了許多關於基因演算法（Genetic Algorithms, GAs）及其應用的知識，以下是我所學到的具體內容，並按照授課的各個單元詳細歸類說明。

CH01: Genetic Algorithms What Are They?

1. Introduction to Genetic Algorithms (GAs):
   基因演算法模仿自然選擇和遺傳的原理，用於解決優化和搜尋問題。它們透過繁殖（Reproduction）、交叉（Crossover）和突變（Mutation）操作來進行全球搜索。

2. The Evolutionary Mechanism:
   Evolution Programs (EPs)、Evolutionary Computing (EC) 和 Evolutionary Algorithms (EAs) 是基因演算法的其他名稱。這些算法依靠遺傳和進化的原理，通過可能解的群體進化來尋找最佳解。

3. Classical Genetic Algorithm:
   Darwin’s Theory: 最強的物種存活並繁殖，從而創造出更優秀的後代。
   Parallel Global Search: GAs 通過同時處理多個候選解來進行全局搜索。
   Operators: 主要包括繁殖（Reproduction）、交叉（Crossover）和突變（Mutation）操作。

4. Key Concepts and Terminologies:
   Population: 一組候選解。
   Genotype (Chromosome): 編碼的解表示。
   Phenotype (Individual): 解的解碼。
   Genes: 解的部分。
   Alleles: 基因的值。
   Locus: 基因在染色體上的位置。
   Fitness: 衡量解好壞的標準。

5. General Scheme of a GA:
   Initialization: 創建初始解群體，通常是隨機的。
   Selection: 選擇最適合的個體進行繁殖。
   Crossover: 結合兩個父代的部分基因以創造後代。

Mutation: 對後代進行隨機變化以保持基因多樣性。
Evaluation: 評估個體的適應度。
Termination: 當達到停止條件時終止算法，如達到最大代數或滿意的適應度水平。

CH02: Genetic Algorithms How They Work?

1. General GA Framework:
Evaluation Function: 確定個體的適應度。
Parameters: 包括群體大小、交叉率、突變率等。
Genetic Representation: 如何編碼解。
Initial Population: 通常隨機生成。
Genetic Operators: 包括選擇、交叉和突變。
Termination Condition: 如達到最大代數或達到滿意的適應度水平。

2. Example of a GA:
多變量優化問題的示例，展示GAs 如何用於解決複雜的優化任務。

3. SGA (Simple Genetic Algorithm):
Initialization: 隨機創建初始群體。
Evaluation: 評估每個個體的適應度。
Selection: 基於適應度選擇個體（如輪盤賭選擇）。
Crossover: 單點交叉結合父代基因。
Mutation: 位翻轉突變以引入多樣性。
Reproduction: 將個體複製到下一代。
Termination: 直到達到停止條件。

4. Genetic Algorithm Procedure:
Begin: 初始化群體並進行評估。
Loop: 直到終止條件，選擇父代，進行交叉和突變，評估新群體。
End: 輸出找到的最佳解。

5. Results and Examples:
提供運行基因演算法在示例問題上的結果，展示解的演化過程。

CH03: Genetic Algorithms Why Do They Work?

1. Schema Theorem:
描述 GAs 通過促進成功的架構（Schema）並消除不成功的架構來工作。
Schemas: 基因值的模式，根據其適應度進行評估。
Order and Defining Length: 階數指固定位置的數量，定義長度是第一個和最後一個固定位置之間的距離。

2. Definitions and Effects on Schema Dynamics:
Effect of Selection: 指數級地促進高於平均水平的架構。
Effect of Crossover: 通過組合不同架構的部分創建新的架構。
Effect of Mutation: 通過改變現有架構引入新架構。
Combined Effects: 選擇、交叉和突變共同引導搜索過程朝向最佳解。

3. Conclusion: Schema Theorem and Building Block Hypothesis:
   Schema Theorem: 確保短的、低階的、高於平均水平的架構得到指數級增加的嘗試。
   Building Block Hypothesis: GAs 通過組合小的高性能架構（構建塊）來形成最佳解。

CH05: Binary or Float?

1. Binary Representation:
   適用於離散領域。
   使用二進制編碼解。
   優點：實現簡單，適用於離散變量問題。

2. Gray Coding:
   二進制數和 Gray Code 之間的轉換方法。
   Procedure: 詳細說明如何將二進制轉換為 Gray Code 以及如何將 Gray Code 轉換為二進制。

3. Floating Point Representation:
   適用於連續領域。
   使用實數編碼解。
   優點：更高的精度，更適合處理約束和設計操作符。

4. Comparison and Challenges:
   Binary vs. Float: 二進制簡單但精度較低；浮點精度高但實現複雜。
   Challenges: 二進制可能無法細緻表示搜索空間，而浮點需要更多計算資源。

CH06: Fine Local Tuning

1. Mutation Techniques:
   Uniform Mutation: 以固定概率獨立改變每個基因。
   Nonuniform Mutation: 允許通過調整突變率來進行微調。

2. Crossover Techniques:
   Simple Crossover: 單點交叉。
   Uniform Crossover: 每個基因隨機選擇自一個父代。
   Arithmetic Crossover: 通過算術運算組合父代基因。

3. Adaptive Probabilities:
   根據當前代的表現調整交叉和突變概率。
   Example: 如果群體適應度停滯，則增加突變率。

CH07: Handling Constraints

1. Constraint Handling Techniques:
   Penalty Functions: 對違反約束的解的適應度進行懲罰。
   Repair Algorithms: 生成解後對其進行修改以符合約束。

2. Hybrid Methods:
   結合多種策略更有效地處理約束。

Example: 使用懲罰函數和修復算法確保解的可行性。

CH08: Evolution Strategies and Other Methods

1. Evolution Strategies (ES):
   1970年代在德國開發，用於數值優化。
   Features: 優化速度快，適用於實數優化，自適應突變參數。

2. Genetic Programming (GP):
   1990年代開發，用於機器學習任務。
   使用樹結構表示解。
   Applications: 預測、分類等機器學習任務。

3. Multimodal and Multiobjective Optimization:
   Multimodal Optimization: 尋找問題中的多個局部最優解。
   Multiobjective Optimization: 同時優化多個相互衝突的目標。
   Methods: Pareto 排序、Niching 方法等用於處理多個目標和解。

1. Introduction to Transportation Problem:
   一個經典的優化問題，涉及從多個供應商向多個消費者分配貨物。
   Objective: 在滿足供應和需求約束的同時最小化運輸成本。

2. Formulation and Examples:
   使用線性規劃進行數學公式化。
   Examples: 展示如何應用運輸問題公式於實際場景。

1. History and Evolution of Programs:
   演化計算方法已經發展出來解決複雜的優化問題。
   Examples: 基因演算法、演化策略和基因程序設計。

2. Applications and Methods:
   Applications: 優化、機器學習和各種工程問題。
   Methods: 不同的演化算法及其特定使用案例。

1. Function Analysis and Exact Solutions:
   Objective: 找到給定函數的最小值和最大值。
   Exact Solutions: 最小值在(0, 0)處為0，最大值在多個點處為2.1913。

2. Method Comparisons and Results:
   Methods: 基因演算法（GA）、爬山法（Hill Climbing, HC）和模擬退火（Simulated Annealing, SA）。
   Parameters: 群體大小、突變率、交叉率等。
   Results: 浮點表示在精度和收斂速度方面通常優於二進制表示。

3. Visualization and Parameter Analysis:
   Visualization: 函數優化的圖形表示。
   Parameter Analysis: 不同參數對基因演算法性能的影響。

1. Stock Prediction Using GA and LSTM:
   結合基因演算法（GA）和長短期記憶網絡（Long ShortTerm Memory, LSTM）進行股票價格預測。
   Objective: 優化特徵選擇和超參數以提高預測精度。

2. Features and Hyperparameters Optimization:
   Features Optimization: 選擇最相關的預測特徵。
   Hyperparameters Optimization: 調整模型參數如學習率、層數等。

3. LSTM Model and GA Flow:
   LSTM: 能夠處理序列和高維數據。
   GA Flow: 描述使用GA優化LSTM模型的過程。

1. Introduction to NHGA for TSP:
   NHGA: 新型混合基因演算法，結合GA和局部搜索啟發式方法，用於旅行推銷員問題（Traveling Salesman Problem, TSP）。
   Objective: 提高解的質量和效率。

2. NHGA Operators:
   Selection Operator: 輪盤賭選擇。
   Crossover Operator: 使用貪婪算法結合父代。
   Mutation Operator: 使用 LinKernighan 啟發式進行突變。
   BestIndividual Set Operator: 保持最佳個體集以保持多樣性。

3. Experiments and Results:
   Comparison: NHGA 與傳統 HGA 的比較。
   Results: NHGA 在解的質量和計算效率方面顯示出改進。

1. Problem 1: Minimum Finding:
   Objective: 使用二進制和浮點表示的GA尋找最小值。
   Parameters: 群體大小、突變率、交叉率等。
   Results: 浮點表示在精度和收斂速度方面表現更好。

2. Problem 2: Maximum Finding:
   Objective: 使用GA、HC和SA尋找最大值。
   Parameters: 與問題1類似，但針對不同方法進行調整。
   Results: GA 總體表現更好，HC 在特定情況下表現優於SA。

3. Method Comparison and Analysis:
   Analysis: 詳細比較GA、HC和SA在精度、收斂速度和計算效率方面的表現。

Conclusion: GA 通常優於HC和SA，浮點表示比二進制更有效。

1. Introduction to Adaptive Disentangled Transformer (ADT):
   ADT: 用於序列推薦任務。
   Objective: 改進學習表示的解開以捕捉多樣化的用戶興趣。

2. Problems and Solutions:
   Problems: 現有 Transformer 架構缺乏層級解開的正規化。
   Solutions: 引入輔助目標，自適應確定最佳解開程度。

3. Methodology and Experimental Results:
   Methodology: 包括重建目標、獨立性目標和自適應訓練策略。
   Experimental Results: 顯示 ADT 在改進推薦性能方面的有效性。

1. Optimization Problems and Pareto Ranking:
   Pareto Ranking: 解決多目標優化問題的方法。
   Objective: 找到代表相互衝突目標間權衡的非支配解集。

2. Examples and Solutions:
   Example 1: 最小化 $f_1(x) = \sin(x) + 1$ 和 $f_2(x) = \cos(x) + 1$。
   Example 2: 最大化 $f_1(x_1, x_2) = x_1 + 3x_2$ 和 $f_2(x_1, x_2) = 3x_1 + x_2$ 受限於約束條件。
   Solutions: 使用 Pareto 排序進行解決並可視化解。

這些學習經驗極大地豐富了我對基因演算法及其應用的理解，並且對於不同優化方法的比較和分析提供了深刻的見解。如有需要，我可以提供更詳細的特定部分分析。

謝謝教授的指導，這些學習將對我未來的研究和實踐有極大的幫助。

在最近的學習過程中，我深入探討了基因演算法（Genetic Algorithms, GAs）及其應用。以下是我所學到的具體內容以及我對這些知識的理解和整理。

基因演算法（GAs）是一種模仿自然選擇和遺傳原理的演算法，用於解決優化和搜尋問題。它們通過繁殖（Reproduction）、交叉（Crossover）和突變（Mutation）操作來進行全局搜索。這些演算法利用遺傳和進化的原理，通過群體的進化來尋找最佳解。經典的基因演算法基於達爾文的理論，認為最強的物種存活並繁殖，從而創造出更優秀的後代。基因演算法通過同時處理多個候選解來進行並行全局搜索，主要操作包括繁殖、交叉和突變。

在學習過程中，我了解了基因演算法的基本概念和術語，包括群體（Population）、基因型（Genotype, Chromosome）、表型（Phenotype, Individual）、基因（Genes）、等位基因（Alleles）、基因座（Locus）和適應度（Fitness）。基因演算法的一般流程包括初始化群體、選擇最適合的個

體進行繁殖、結合兩個父代的部分基因以創造後代、對後代進行隨機變化以保持基因多樣性、評估個體的適應度，最後在達到停止條件時終止演算法，如達到最大代數或滿意的適應度水平。

基因演算法的框架包括評估函數（Evaluation Function）、參數設置（Parameters）、遺傳表示（Genetic Representation）、初始群體（Initial Population）、遺傳操作符（Genetic Operators）和終止條件（Termination Condition）。這些操作符包括選擇（Selection）、交叉（Crossover）和突變（Mutation），並且在達到最大代數或滿意的適應度水平時終止。

在實際應用中，我學習了一些基因演算法的示例，例如多變量優化問題，展示了GAs如何用於解決複雜的優化任務。簡單基因演算法（Simple Genetic Algorithm, SGA）包括初始化、評估、選擇、交叉、突變和繁殖的步驟，直到達到停止條件。通過這些示例和實驗結果，我了解了基因演算法在解決實際問題中的效果，展示了解的演化過程。

在學習過程中，我還探討了為什麼基因演算法有效的原因。架構定理（Schema Theorem）描述了GAs如何通過促進成功的架構（Schema）並消除不成功的架構來工作。架構是基因值的模式，根據其適應度進行評估。架構的階數（Order）指固定位置的數量，定義長度（Defining Length）是第一個和最後一個固定位置之間的距離。選擇、交叉和突變的結合效果引導了搜索過程朝向最佳解。

在學習二進制表示和浮點表示時，我了解到二進制表示適用於離散領域，而浮點表示適用於連續領域。二進制表示使用二進制編碼解，實現簡單，適用於離散變量問題。格雷碼（Gray Coding）是二進制數和格雷碼之間的轉換方法。浮點表示使用實數編碼解，具有更高的精度，更適合處理約束和設計操作符。二進制和浮點表示各有優缺點，二進制簡單但精度較低，浮點精度高但實現複雜。

在進一步研究中，我學習了精細局部調整的方法。突變技術包括均勻突變（Uniform Mutation）和非均勻突變（Nonuniform Mutation），允許通過調整突變率來進行微調。交叉技術包括簡單交叉（Simple Crossover）、均勻交叉（Uniform Crossover）和算術交叉（Arithmetic Crossover）。自適應概率（Adaptive Probabilities）根據當前代的表現調整交叉和突變概率，例如在群體適應度停滯時增加突變率。

在處理約束方面，我學習了幾種處理約束的方法，包括懲罰函數（Penalty Functions）和修復算法（Repair Algorithms）。懲罰函數對違反約束的解的適應度進行懲罰，而修復算法則生成解後對其進行修改以符合約束。混合方法結合多種策略更有效地處理約束，如使用懲罰函數和修復算法確保解的可行性。

此外，我還學習了演化策略（Evolution Strategies, ES）和其他方法。演化策略在1970年代在德國開發，用於數值優化，具有優化速度快，適用於實數優化，自適應突變參數的特點。基因程序設計（Genetic Programming, GP）在1990年代開發，用於機器學習任務，使用樹結構表示解。這些方法應用於預測、分類等機器學習任務。多模態優化（Multimodal Optimization）尋找問題中的多個局部最優解，而多目標優化（Multiobjective Optimization）則同時優化多個相互衝突的目標，使用 Pareto 排序和 Niching 方法等技術處理多個目標和解。

在運輸問題的研究中，我學習了一個經典的優化問題，涉及從多個供應商向多個消費者分配貨物。運輸問題的目標是在滿足供應和需求約束的同時最小化運輸成本。我學會了如何使用線性規劃進行數學公式化，並通過實際示例展示如何應用運輸問題公式於實際場景。

此外，我還學習了基因演算法的歷史和程序演化，了解了演化計算方法的發展和應用，例如基因演算法、演化策略和基因程序設計。這些方法廣泛應用於優化、機器學習和各種工程問題，具有不同的使用案例和應用場景。

在具體實驗和應用中，我學習了如何使用基因演算法進行函數分析和精確解，包括找到給定函數的最小值和最大值。通過比較基因演算法（GA）、爬山法（Hill Climbing, HC）和模擬退火（Simulated Annealing, SA），我了解了不同方法的性能差異，並分析了不同參數對基因演算法性能的影響。

總結這段學習經歷，基因演算法及其應用的研究使我對優化方法有了更深入的理解，並提供了實際問題解決的工具和技術。這些知識和技能將對我未來的研究和實踐提供極大的幫助。

感謝教授願意給予機會學習

深度學習網路基礎: 理論與產業實務
FUNDAMENTALS OF DEEP LEARNING NETWORKS: THEORY AND INDUSTRIAL APPLICATIONS

開課教授:陳朝鈞 Chen, ChaoChun

在這段時間裡，我完成了多個程式專案，這些程式專案極大地豐富了我的知識儲備並提升了我的實踐能力。

首先，我完成了一項關於大型語言模型（LLM）和API開發的程式專案。這次程式專案的主要目標是搭建一個LLM模型並開發一個API伺服器，允許使用者通過API對LLM模型進行Prompt並獲得Response。在這過程中，我使用了PyTorch框架來開發LLM模型，該模型能夠接受Prompt文字並生成相應的回應。API伺服器則使用Django框架開發，能夠通過HTTP方式接收請求並返回結果。為了達成這一目標，我撰寫了一份詳細的操作手冊，說明如何搭建和測試系統，並將系統封裝在Docker容器中，方便測試和部署。此外，我還提交了一份測試報告，詳細描述了系統的執行結果。在程式碼開發方面，我嚴格遵守命名規範，保持程式碼的重用性，並將程式碼結構化管理，確保了系統的可讀性和維護性。

接下來，我參與了一個關於減少Transformer模型CPU和記憶體負載的程式專案。在這項程式專案中，我的任務是優化Transformer模型，以減少其在訓練和推理過程中的CPU和記憶體負載。我首先建立了一個Transformer模型，並使用該模型預測台積電（2330）的明日股價。為了提升模型的效能，我採用了Mamba架構，並對模型進行了調優。隨後，我對比了優化前後模型的大小、準確率和推論效率，通過回測方法評估模型的表現。為了確保程式專案的完整性，我撰寫了一份操作手冊，詳細說明了如何優化模型和執行推論，並提交了包含原始模型和優化後模型的程式碼和模型文件。

在進行模擬股票交易系統的期末專題中，我學習了如何開發和操作一個模擬股票交易系統。這個系統包括多項功能，如小組登入/登出、查詢個股歷史股價和最新股價、瀏覽和下載交易紀錄以及查看最新公告和資產總額排行榜。系統在特定時間開放用戶查看交易績效及進行程式下單，並在每日開盤後更新資產總額排名。在這項專題中，我不僅開發了一個深度學習模型

來預測股市走勢，還撰寫了一份詳細的報告，並錄製了一段演示視頻，展示系統的功能和模型的預測結果。

另外，我完成了一個設計自動股票交易機器人的專案項目。這個項目旨在設計一個自動交易的股票機器人，根據使用者設定的條件進行交易。為了達到這個目的，我使用Python程式進行選股，並整合到買賣程式中。我選擇了高獲利率的選股方程式，以提高獲利機率。每張股票有不同的性質，因此我根據其性質分類採取不同的交易策略，包括設定機器人的交易動作（如買入、觀望、賣出），導入短期操作看盤技巧，並運用AI預測股價以提升交易獲利。

在選股方面，我採用了Beautifulsoup（bs4）進行網路爬蟲，並使用Talib進行技術分析。我尋找了融資融卷大增的股票，特定券商買進賣出的股票，以及月營收和年營收增加的績優股，並對這些股票進行排序。同時，我也關注了技術線型呈現黃金交叉的股票，並尋找出超跌、止跌回升的股票。

在交易過程中，我根據RSI指標和技術線型來決定買進、觀望和賣出的時機。例如，當RSI<=25且出現黃金交叉，AI預測明日上漲，買盤湧入時，我會考慮買進；當RSI介於25到80之間且內外盤差未擴大，技術線型沒有跌破時，我會觀望；當RSI>=80且大量賣單浮現，融資暴增，外資賣出時，我會賣出。

此外，我還分析了不同交易策略的獲利程度，並比較了人為操作與機器人操作的差異。我使用openpyxl在每次交易時記錄買進或賣出的操作紀錄及其背後的邏輯，方便後續改善。

然而，我在這個項目中也遇到了一些挑戰，例如虛擬股市系統無法超額下單，每間上市公司公告股東會日期不固定，以及AI演算法預測股價需要較多硬體資源且費時。此外，如何縮小掃描範圍，加速掃描週期，並不斷更新買賣點資訊也是需要克服的問題。

最後，我參與了一個關於Prompt Engineering的程式專案，在通過不同的Prompt Engineering技術來增強CLIPseg模型的影像切割準確度。在這項專案中，我首先使用CLIPseg模型並應用不同的Prompt Engineering技術，包括通用提示模板、提示集成、負向提示和ChainofThought提示，來增強模型在Pascal5i資料集上的表現。隨後，我將改進後的CLIPseg模型應用於個人照片的影像切割，並提交了切割後的可視化影像。這些技術使模型在不進行微調的情況下，顯著提高了影像切割的準確度。為了展示我的研究成果，我撰寫了一份實驗比較結果報告，並提交了包含Prompt Engineering設計描述的簡報。

通過這些程式專案的完成，我加深了對大型語言模型、深度學習框架及API開發的理解，掌握了如何優化模型性能以及進行Prompt Engineering的技巧。

*****

首先，我對Vision-Language Models (VLMs) 及其代表性模型CLIP有了深入的了解。VLM是一種多模態模型，能夠同時處理視覺和文本信息。這種模型的應用範圍非常廣泛，包括圖像標註、文本到圖像生成以及基於圖像的搜索。CLIP作為一個預訓練的VLM模型，能夠在沒有直接優化的情況下執行各種視覺任務，這主要得益於它將大量的圖像-標註對數據進行訓練，從而建立起文本描述與視覺特徵之間的聯繫。在學習過程中，我了解了如何設計和優化提示詞（

prompt)，並掌握了如Prompt Ensemble和CoOp（Context Optimization）等技術，以提高CLIP模型在下游任務中的性能。

此外，我還研究了基於CLIP的異常檢測技術。異常檢測在許多領域都有重要應用，例如信用卡欺詐檢測、工業損傷檢測和醫療成像。通過學習，我掌握了如何利用WinCLIP模型來進行異常檢測。WinCLIP使用正常和異常提示詞來計算得分圖，以此來識別圖像中的異常點。這一方法在應對非獨立同分佈數據時表現出色，特別是在少樣本學習場景中，異常檢測的準確性得到了顯著提高。

在聯邦學習領域，我學習了如何在分散式設備上協作訓練模型，而不需要交換數據。這種方法能夠有效保護用戶隱私，並減少數據傳輸的成本。然而，聯邦學習面臨的主要挑戰之一是數據的非獨立同分佈問題。為了解決這一問題，我研究了多種方法，包括FedProx、Model-Contrastive Federated Learning (MOON) 和FedDC。這些方法通過不同的技術手段，調整本地和全局模型之間的權重差異，從而提高了聯邦學習模型的收斂速度和準確性。

Transformer模型的學習是我此次研究的另一個重要方面。傳統的RNN模型在處理長序列數據時，常常面臨難以捕捉長距依賴關係和缺乏並行計算能力的問題。Transformer模型通過引入自注意力機制，成功解決了這些問題。自注意力機制允許模型直接計算序列中任意兩個元素之間的關係，從而更好地捕捉長距依賴。同時，Transformer的結構允許並行計算，極大地提高了計算效率。在此基礎上，我還學習了Vision Transformer (ViT)，該模型將圖像處理視為序列數據處理，並在大數據預訓練的情況下，表現優於傳統的卷積神經網絡（CNN）。

在模型訓練過程中，我深入探討了如何處理過擬合和欠擬合問題。過擬合指的是模型在訓練數據上表現良好，但在未見數據上表現不佳，而欠擬合則是模型在訓練數據和未見數據上均表現不佳的情況。為了解決這些問題，我學習了多種正則化技術，包括L1正則化、L2正則化和Dropout。這些技術通過增加損失函數中的正則化項，限制模型過於複雜，從而提高模型的泛化能力。

總結來說，通過這段學習，我對深度學習中的多個核心概念和技術有了全面的理解，並掌握了如何應用這些技術來解決實際問題。

生物與醫學資訊專論
SPECIAL TOPICS ON BIOINFORMATICS & BIOMEDICAL INFORMATICS
開課教授: 賀保羅 PAUL BRICE HORTON Ⅱ

在這門課程中，我深入學習了計算生物學和醫學資訊學的多方面內容，並且通過多篇重要的研究論文加深了對這些領域的理解。

首先，我學習並分析了幾篇極具影響力的研究論文。其中，Stephen F. Altschul等人在《Journal of Molecular Biology》上發表的《Basic Local Alignment Search Tool》介紹了BLAST工具，該工

具通過快速的序列比對演算法，顯著提高了基因和蛋白質序列分析的效率。這一工具的出現不僅在生物信息學中產生了深遠的影響，也為後續的研究提供了強有力的技術支持。

在課程中，我還研究了K. Lari和S. J. Young在《Computer Speech and Language》上發表的兩篇論文《The Estimation of Stochastic Context-Free Grammars Using the Inside-Outside Algorithm》和《Applications of Stochastic Context-Free Grammars Using the Inside-Outside Algorithm》。這些文章詳細探討了內外演算法在估計隨機上下文無關文法中的應用，並展示了該方法在語音識別中的實踐成果。這些研究使我對隨機文法和語音識別技術有了更深刻的理解，並啟發我思考這些技術在生物信息學中的潛在應用。

此外，我也閱讀了Sean R. Eddy在《Bioinformatics》上發表的《Profile Hidden Markov Models》。這篇文章深入介紹了隱馬爾可夫模型在蛋白質結構預測中的應用，並探討了這些模型在處理生物序列數據時的強大能力。通過學習這篇文章，我不僅掌握了隱馬爾可夫模型的基本理論，還理解了其在實際應用中的重要性。

在學習期間，我還接觸到K. Sjölander等人在《Computer Applications in the Biosciences》上發表的《Dirichlet Mixtures: A Method for Improved Detection of Weak but Significant Protein Sequence Homology》。這篇文章提出了一種基於狄利克雷混合模型的方法，用於改善弱但顯著的蛋白質序列同源性檢測。這一研究為我展示了如何利用統計模型來提高生物序列分析的準確性和敏感性。

在課堂上，我積極參與文獻展示和討論。每次展示前，我都會進行充分準備，確保能夠清晰地解釋每篇論文的核心內容和研究方法。在討論中，我學到了如何批判性地看待研究成果，並從不同的角度思考問題。這些經驗不僅提升了我的表達能力，也增強了我的批判性思維。

作為期末項目的一部分，我撰寫了一篇小型評論，連結了多篇在課程中閱讀的論文。在評論中，我探討了這些研究之間的共通點和差異，並提出了自己的見解。例如，我比較了BLAST和隱馬爾可夫模型在序列分析中的不同應用場景和優缺點，總結了它們在實際研究中的影響和應用前景。這一過程不僅使我對所學知識進行了系統性總結，也提升了我撰寫學術評論的能力。

除了理論學習外，我還將所學方法應用到實際數據分析中。例如，我使用BLAST工具對實際的DNA序列進行比對，並嘗試構建隱馬爾可夫模型來預測蛋白質二級結構。這些實踐活動讓我對這些工具的操作有了更深入的理解，也使我認識到理論與實踐相結合的重要性。

此外，我在分析和理解一篇題為《Neural Machine Translation by Jointly Learning to Align and Translate》的報告中，學到了神經機器翻譯（NMT）系統如何通過聯合學習對齊和翻譯來改進翻譯效果。這篇文章由Dzmitry Bahdanau、Kyunghyun Cho和Yoshua Bengio撰寫，介紹了在NMT中引入注意力機制的必要性和優勢。通過引入注意力機制，NMT模型能夠在每一步生成輸出詞彙時動態地專注於輸入序列的不同部分，這顯著提升了翻譯的準確性和上下文相關性。

這個報告深入解釋了基本的編碼-解碼架構、注意力機制的工作原理以及雙向RNN的應用。實驗結果表明，該模型在處理長句子時表現尤為出色，這是因為注意力機制能夠幫助模型保留並訪問整個句子的相關信息，而不僅僅依賴於壓縮的固定大小向量。這一發現強調了注意力機制在現代NMT系統中的重要性，使其能夠在準確性和上下文敏感性方面超越傳統模型。

在另一篇報告《How to Apply de Bruijn Graphs to Genome Assembly》中，我學習了如何將de Bruijn圖應用於基因組裝。這篇文章由Compeau, Pevzner和Tesler撰寫，展示了如何利用de

Bruijn圖將基因組裝問題轉化為圖遍歷問題，從而使得處理大規模序列數據成為可能。該方法通過將基因組裝從尋找哈密頓環轉變為尋找歐拉環，顯著提高了計算效率，並且能夠有效處理來自下一代測序技術的大規模數據集。

這篇報告介紹了de Bruijn圖的構建方法，包括如何將(k-1)-mers作為節點、k-mers作為邊來構建圖，並使用歐拉演算法來尋找圖中的循環。此外，文章還討論了如何處理測序錯誤和DNA重複問題，從而提高組裝的準確性和可靠性。這些方法展示了de Bruijn圖在基因組裝中的強大應用前景，並啟示了在處理複雜基因組時需要考慮的挑戰和解決方案。

總之，這門課程讓我對計算生物學和醫學資訊學有了更深刻的理解。我不僅學會了許多重要的研究方法，還提升了文獻分析和批判性思維的能力。

在本學期的異常檢測課程中，我有幸參與了一系列挑戰性極高程式專案，這些程式專案不僅使我加深了對深度學習模型的理解，還提升了我的實作和應用能力。以下是我對這些程式專案的詳細學習報告，涵蓋了各種不同模型和算法的實作設計及其結果。

程式專案目標與背景：

我們的主要目標是使用MNIST手寫數字資料集進行異常檢測，將數字「1」、「3」、「5」、「7」視為正常類別，其餘數字視為異常類別。程式專案涵蓋了Convolutional Neural Network (CNN) 分類器、Auto Encoder、Denoising Auto Encoder、Variational AutoEncoder (VAE)、Isolation Forest方法，以及結合預訓練模型與Isolation Forest的方法。

CNN分類器的實作與結果

在這次程式專案中，我設計並訓練了一個卷積神經網絡（CNN）來進行異常檢測。模型包含兩層卷積層，每層後接最大池化（max-pooling）和dropout層，接著是兩層全連接層，最終輸出兩個類別：Normal和Anomaly。訓練過程中，我使用了交叉熵損失函數（CrossEntropyLoss）和Adam優化器。訓練數據集包含50000筆數據，測試數據集10000筆，進一步分為5000筆作為測試集，5000筆作為驗證集。最終，模型在測試集上的最高準確率達到99.4%，在驗證集上的準確率為99.3%。混淆矩陣顯示了模型在分類正常和異常數字上的高準確性。

Auto Encoder的實作與結果

接下來，我實作了一個自編碼器（Auto Encoder），該模型由編碼器和解碼器組成，均包含卷積層。在訓練過程中，我僅使用正常類別的數據，並選擇均方誤差（MSE）作為損失函數。雖然模型在重建正常數據上表現良好，但在異常檢測上的驗證準確率僅為42.56%。混淆矩陣顯示了Auto Encoder在區分正常和異常數字上的困難。

Denoising Auto Encoder的實作與結果

然後，我設計了一個去噪自編碼器（Denoising Auto Encoder），其模型結構與Auto Encoder相同，但在訓練過程中對輸入數據添加了標準差為0.1的高斯噪聲。這種方法的目的是讓模型學習到更穩健的特徵。然而，去噪自編碼器的驗證準確率為42.52%，與標準自編碼器相似，顯示在異常檢測上的效果仍不理想。

Variational AutoEncoder (VAE)的實作與結果

隨後，我實作了一個變分自編碼器（VAE），該模型的編碼器包含卷積層和計算均值及對數方差的全連接層，而解碼器由卷積反轉層（convolutional transpose layers）組成。訓練過程中，我使用二元交叉熵（BCE）和KL散度（KL Divergence）作為損失函數。VAE在異常檢測上的驗證準確率為40.58%，顯示該模型在此任務上的表現同樣不理想。

Isolation Forest方法的實作與結果

我還嘗試了使用Isolation Forest方法進行異常檢測。這種方法基於從CNN模型中提取的特徵進行訓練。在實驗中，Isolation Forest在驗證集上的準確率為49.43%，雖然比自編碼器和VAE略有改善，但仍未能達到理想效果。

預訓練模型與Isolation Forest結合的方法

最後，我結合了預訓練的CNN模型和Isolation Forest進行異常檢測。這種方法先使用CNN模型提取特徵，然後再將這些特徵用於訓練Isolation Forest。結果顯示，這種方法的驗證準確率為48.18%，與單獨使用Isolation Forest的性能相近。

程式專案**1**：全連接層神經網絡的設計與應用

在程式專案1中，我使用`make_circles`生成了100個二維、兩類的數據點，並將這些數據分成70%的訓練集和30%的測試集。我設計了一個僅包含全連接層的神經網絡來進行分類。模型的訓練過程使用了PyTorch進行實作，並在Jupyter Notebook中完成，提交了ipynb文件。

程式專案**2**：深度神經網絡在手寫數字識別中的應用

程式專案2要求我們使用解析度為128x128的手寫數字圖片進行分類。我設計了一個深度神經網絡，訓練該模型來預測圖片中的數字。訓練過程中生成了混淆矩陣，計算了top-1和top-3的準確率，並提交了包含預測結果的CSV文件。

程式專案**3**：捲積神經網絡在振動信號分類中的應用

在程式專案3中，我設計了一個捲積神經網絡（CNN）來分類軸承振動信號。數據集包含正常、外部斷裂和內部斷裂的振動信號波形。我將這些波形分段處理，以降低模型的複雜度。訓練完模型後，我使用測試集進行推論，並將結果記錄在CSV文件中。

程式專案**4**：循環神經網絡（**RNN**）在圖片分類中的應用

程式專案4要求我們使用RNN模型來分類手寫數字圖片。我比較了不同類型的RNN層，如SimpleRNN、LSTM和GRU，並比較了它們的訓練結果。我還使用PyTorch實作了一個模型，生成類似於給定數據分佈的數據，並在圖表中顯示這些數據。

程式專案**5**：多種異常檢測方法的比較

在程式專案5中，我實作並比較了多種異常檢測方法，包括標準AutoEncoder、去噪AutoEncoder、變分AutoEncoder（VAE）、Isolation Forest以及結合預訓練模型與Isolation

Forest的方法。我詳細比較了這些方法在異常檢測任務中的性能，並根據不同的方法設置了合適的閾值。

程式專案6：物體預測系統的設計與應用

程式專案6要求我們設計一個基於深度神經網絡的物體預測系統，來預測測試圖片中數字的類別和邊界框座標。我設計並訓練了一個物體檢測模型，並生成了包含預測結果的CSV文件。此外，我還使用了普通GAN、WGAN和Info-GAN來生成數據，並控制生成數據的類別。

額外實作與結果

Wasserstein GAN（WGAN）的實作與結果

在WGAN實作中，我設計了生成器（Generator）和鑑別器（Discriminator）神經網絡，並使用Gradient Penalty進行訓練。生成數據與真實數據的分佈圖顯示，生成器能夠生成類似於真實數據的數據點。

Info-GAN的實作與結果

我還實作了一個Info-GAN模型，包括InfoGAN_Generator和InfoGAN_Discriminator，並包含一個Q網絡，用於學習解耦表示（disentangled representation）。這種模型能夠控制生成數據的類別，並在圖表中可視化生成數據。

視覺化結果

在每個實驗中，我使用混淆矩陣顯示了模型在驗證集上的分類性能，並在圖表中展示了生成數據的分佈，與真實數據進行比較。

這些詳細的分析和結果展示了不同異常檢測模型和技術的優缺點。CNN分類器在本次任務中表現最佳，而基於GAN的模型則提供了生成建模能力

總結與反思

整體來說，CNN分類器在本次程式專案中的表現最為優異，驗證準確率達到99.3%。相比之下，Auto Encoder、Denoising Auto Encoder、VAE和Isolation Forest在異常檢測上的表現相對較差，驗證準確率約在40-50%之間。這表明，在處理類別數量眾多且資料簡單的任務時，CNN分類器的效果更佳。

此外，我還進行了Wasserstein GAN（WGAN）和InfoGAN的實作，這些模型展示了生成建模的能力，並進一步提升了我對生成對抗網絡（GAN）的理解。

感謝教授給予機會讓我加入課程學習，這些程式專案更讓我在異常檢測和深度學習領域獲益匪淺。不僅提升了我的理論知識，也增強了實際應用的能力。希望未來能有更多機會繼續深入學習和研究。