## 自傳

## 學士前

最早接觸程式相關的時候是**國小三年級**,那是 Yahoo 還可以與 Google 一決高下、掀蓋式手機還是流行的時候。第一 本有關於程式設計的書名是【如何學好程式設計】,裡面的內容放到如今雖然已經過時,但是我仍記得裡面的許多概 念,例如:編譯器、直譯器等等。那時候我就會參考書上的程式碼,並且在非常舊的筆記型電腦上試著執行,但那時 缺乏在網路上查找軟體的能力,即使我知道書上告訴我要先編譯、連結,才能輸出可用的執行檔,我也無法順利運行 我的第一支 "Hello World",但這不妨礙我對於程式語言的熱忱。我的第二本書則是【C語言從零開始】,其內容與如 今的 C 語言教學本類似,不過當時我並沒有"變數"、"賦值"的概念 (小學三年級甚至還沒教方程式),所以理解其中 的演算法十分困難,我為了搞懂"氣泡排序法"就花了非常多時間,至少一個禮拜。而在某天下午,我突然意識到 C 語言中的等號其實是"複製一份值到指定變數",否則這個演算法就無法成立,從那時候開始,我的學習程式語言的速 度突飛猛進,即使沒有電腦可以"執行"那些程式,我也可以在心中模擬那些程式。在小學六年級時,我參加了有關 Arduino 的課程,我發現裡面的程式與 C 語言非常相似,於是跑去問了教學的老師——侯凱喨—能不能教我 C 語言,老 師只是笑著說看之後有沒有時間,但之後就沒有相關課程了。我從主任那裡聽到他在某間私校任教,我當時一股腦的 與家長說,我要去讀那間學校,因為家境原因,起初家長十分反對,但所幸我在入學考時,取得相對優秀的成績,因 而有獎學金減免,家長才答應讓我去讀那間私校。進入私校後,我開始向侯老師學習,在這段期間我每天中午都會去 電腦教室寫程式,**我很把握每一次碰到電腦的機會**,因為其他時間我碰不到電腦。在國中期間,我接觸了許多基礎的 演算法以及資料結構,例如:BFS、DFS、Dijkstra、Linked List 等等,並且熟悉 C++ 的基礎物件導向語法。在國三左 右,在另外一位老師—陳揮明—的推薦下,我學習 LEGO 自走車,並且參加了不分年級的比賽(自走車尋跡+避障), 拿到第三名的成績。在此期間還學到了比例-積分-微分控制器 (PID controller) 的概念,並應用在自走車上。

高中時,因為那時候想要在學科能力競賽取得好成績,然後透過特殊選材的管道進入大學,我主動在電腦課上向林品杰老師詢問能不能參加競賽程式,由於嘉義高中參加競賽程式的人寥寥可數 (包括我 5 個人,且其餘都是學長),所以老師在測試我的能力後就爽快的答應了。在高中這三年,我參加過許多競賽程式,但都無法拿下好成績,即使學的東西比起以往還要深入、廣泛,但貌似我的對手都遠遠比我還要強大,解題的速度、想法都是我無法比擬的,高中算是我的低潮期。在這段期間,我偶爾會去寫寫課堂外我發現很有趣的東西,大部分與數學相關,並放在我個人的網頁上,有時只是很簡單的小發現,例如:如果將化學課教的體心立方結構推廣成高維度,裡面球的體積會超過容納它的正方體的體積,而有時候會是非常難的東西:階乘推導至 Gamma 函數的過程、最速降線的推導等等,我那時也尚未意識到,或許這才是我想做的,也就是做研究。

## 學士班一年級 到 二年級

進入中山大學後,由於我在高中時期有競賽相關的經歷,因此在大學一年級時,就獲得了 C 程式設計等基礎課程的抵免資格,我利用多出來的空堂修習高年級的課程。同時,代表了學校參與全國大專生軟體設計競賽以及 TOPC 台灣線上程式設計競賽,且成功取得校內第一,代表中山參加 ICPC 亞洲桃園區程式競賽。同時在大一時,我修習高等物件導向程式設計,在這堂課中我學習到了與以往競賽程式不同的「程式」,例如 C/C++ 執行時的記憶體配置、Metaprogramming、C++ 如何處理函式多載、虛擬建構子的用途、類別推導等許多非常新奇且需要看大量的文件才能知道的知識。這讓我知道其實在競賽程式之外,還有許多非常有趣的領域。在大學一年級時,我就認識了一些學長,而這些學長也成了後來我加入程式研習社的契機。我在程式研習社中主要負責教學組,由於我有非常驚人的 C/C++ 使用資歷,我負責關於進階 C/C++ 的課程。而同時,我成為蔡崇煒副教授的專題生,我一開始先參加學長們與教授的會議,了解目前實驗室的研究領域,同時學習進度把控、報論文的技巧及如何撰寫論文等研究生必備的技能。

在這之後,我對於無訓練神經網路架構搜尋 (Training-Free NAS) 演算法有研究興趣與新想法,提出了一篇研究論文並成功投稿國際論文的研討會,在這篇論文中,我利用排名的方式結合異質性高的評估函數,同時利用退火模擬演算法幫助搜尋最佳的神經網路架構。同時我也負責撰寫另一篇學姊的論文,以此打磨自己的論文撰寫技巧,在這篇論文中,我們使用新的超啟發式演算法,Search Economics,找到最佳的方法減枝神經網路,並在之後使用知識蒸餾修復此神經網路的表現。在此期間我逐漸熟悉了 PyTorch、Scikit-learn 等機器學習相關套件,且使用其完成演算法設計。

## 學士班三年級 到 目前

在成功投稿論文後,我才開始製作專題,這時剛好教授有接到教育部的【青少年網路內容防護計畫】,順水推舟的,我的專題就成了不當資訊防護系統,這個系統應用於學術網路中,從原有的黑名單中,學習其中的特徵,再自動從流量中找到未在黑名單中的不當網站,以此擴增原本較小的黑名單。因為只有一個人,故在時間安排上需要非常精準且有效率,得益於我在二年級的磨練,我可以獨自實作想法、完成論文或報告的撰寫。同時,我也沒有選擇荒廢課業,我在三年級時,資工系上的必修與選修都拿到 A+、專題仍按照進度進行、並且同時參加 AIS3 的選拔比賽,獲得參加資格。在大四時,我參加了各式各樣的比賽並且取得相當的成績,例如:全國大專生軟體設計競賽佳作、ICPC 亞洲台中區程式競賽銅牌、IMBD 2024 全國智慧製造大數據分析競賽進入決賽(尚未開始)、國立中山大學工學院聯合專題競賽佳作。同時,我與另一位碩士學長共同合作撰寫一篇論文,作為我專題的延伸,我們提出了使用 TAIDE 作為資料擴增的方法,我們會先使用一些文句微調 TAIDE 模型,再讓 TAIDE 產生出大綱,之後再用生成的大綱、短語、長句生成目標資料。此論文已經於 2024 ICEA 國際研討會通過審查。

我在小時候總是朝著「我想寫程式」這個方向不斷努力,而隨著我的年紀增長,我在這條路上從來沒有後悔過, 反而這條路越來越明顯,我的方向越來越明確,曾經我聽過這麼一段話:「所謂的天賦,不是那些做的比別人更好的 事,而是那些你不由自主會去做的事」,現在我相信在電腦科學這片汪洋中探索未知的疆界,就是**我的天賦**。