# Berkeley 針對 AI 系統挑戰的報告*

Ion Stoica、Dawn Song、Raluca Ada Popa、David Patterson、Michael W. Mahoney、Randy Katz、Anthony D. Joseph、Michael Jordan、Joseph M. Hellerstein、Joseph Gonzalez、Ken Goldberg、Ali Ghodsi、David Culler、Pieter Abbeel

Translation

Kao, Sheng Chieh
b06902117@ntu.edu.tw
Nation Taiwan University
Taipei, Taiwan

## 摘要

隨著電腦視覺、語音辨識和機器翻譯系統的日益商品化以及機器學習後端技術(如數位廣告和智慧基礎設施)的廣泛設置，人工智慧(AI)已經從實驗室轉向量產。這些變化是由前所未有的資料和計算水平、機器學習的進步、系統軟體和架構的創新以及這些技術的廣泛可用性(broad accessibility)所實現的。

下一代的 AI 系統可望加速這些發展，並透過頻繁地互動以及替人們做出高度客製化的(常常是任務關鍵型)決策來影響我們的生活。然而，要實現這樣的想像會帶來嚴峻的挑戰，特別是，我們需要的是能夠在不可預測的環境中做出及時和安全決策的 AI 系統，這些系統必須能夠對抗強大的攻擊者，並且可以在不侵害機密的條件下，穩健地處理日益增加的(跨組織和個人的)資料。摩爾定律的結束限制這些技術可以存儲和處理的資料量，更加劇了這些挑戰。在本文中，我們將分別針對系統、架構和安全三個方面，提出幾個開放研究方向來面對這些挑戰，並幫助發掘人工智慧改善生活和社會的潛力。

## 關鍵字

人工智慧、機器學習、系統、安全

## 1 介紹

早在 1960 年代，人工智慧就已經發展成一門廣泛使用的工程學科，這門學科是將演算法和資料匯集在一起，以解決各種模型辨識、學習和決策問題，人工智慧也與越來越多的工程和科學研究領域交流，同時影響許多和計算有關的領域。

電腦系統(computer system)已經被證明是近年來催化人工智慧發展的必要條件。平行硬體和可擴展軟體系統的進步推動了新機器學習框架和演算法的發展，讓人工智慧可以解決大規模的現實世界問題。快速降低的存儲成本、眾包(crowd-source)、手機應用、物聯網(IoT)以及資料的強大競爭力(資料就是力量)驅使人們進一步投資資料處理系統和 AI 技術。這些因素讓人工智慧在許多實際任務中有接近(甚至常常超越)人類的能力。成熟的 AI 技術不僅推動了現有行業的發展(例如：網路搜尋、高速的交易或商業行為)，還推動了新興產業的崛起(例如：物聯網、增強現實(AR)、生物技術和自駕車等)。

許多應用都需要 AI 系統透過自主決策與現實世界進行互動，像是自動無人機、機器人手術、醫療診斷和治療、虛擬助手等等。隨著現實世界不斷變化，有時出乎意料，這些 AI 應用需要具備終身學習(life-long learning)和無止境學習(never-ending learning)的能力：終身學習系統能有效地將過去學習的知識轉移並運用到新的任務，同時盡可能地降低災難遺忘(catastrophic forgetting)的影響來依序解決多項任務；而無止境學習是指 AI 系統會掌握一個不斷增長的任務集合，並且於每次學習持續改善集合內所有任務的效能。

滿足這些需求會帶來嚴峻的挑戰，例如：在動態環境中主動進行探索、做出安全可靠的決策(可能有駭客、吵雜(noisy)或無法預期的輸入)、可解釋自身的決策、簡化建立應用的新模組化架構。此外，隨著摩爾定律的結束，人們不能指望計算和存儲的快速增長來解決下一代 AI 系統的問題。

解決這些挑戰需要在架構、軟體和演算法等方面協同進行創新。本文並非討論特定的 AI 演算法或技術，而是研究系統在解決人工智慧挑戰中能擔任的重要角色，並提出一些有前景的研究方向。

## 2 是什麼促成了 AI 近年來的突破

人工智慧近年來的突破是由過去二十年來的「完美風暴」所實現的：(1) 巨量資料、(2) 可擴展的電腦和軟體系統、 (3) 上述技術的廣泛可用性。這些趨勢使我們得以用前所未有的規模和範圍探索人工智慧在各個問題領域中的演算法和架構，例如：深度學習、強化學習和 Bayesian 推理。



*** “A Berkeley View of Systems Challenges for AI”, Technical Report No. UCB/EECS-2017-159, http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-159.html, October 16, 2017.

### 2.1 大數據

隨著線上服務、智慧手機和 GPS 的廣泛使用，Google、亞馬遜、微軟和雅虎等網路公司開始以影音、文字檔和使用者日誌等形式收集大量資料。當機器學習演算法與這些大數據結合使用時，就能在各種核心服務獲得品質更好的結果，例如：資訊檢索、資訊提取和廣告。

### 2.2 大系統

大數據進一步促進了電腦和軟體系統的快速創新。為了能夠存儲大數據，網路服務公司開始建立大規模的資料中心，有些資料中心甚至擁有將近一萬台的伺服器，並提供 EB 級的存儲空間；而為了能夠處理大數據，這些公司也建立了能夠在廉價量產的伺服器叢集上運行的大型軟體系統，像是 Google 開發的 MapReduce 和 Google 文件系統，還有在那之後出現的開源 Apache Hadoop。

許多大幅提高速度、規模和易於操作的系統接著問世，這些硬體和軟體的創新使資料中心成為新的電腦，研究人員和相關業者在這些系統上建立了程式庫，以滿足對機器學習(ML)日益成長的需求。

近年來，深度學習(DL)的成功也催生了新一波的軟體系統(例如：TensorFlow、Caffe、Chainer、PyTorch 和 MXNet)，這些軟體系統讓計算工作可以分散到 CPU 叢集，並有效地運用專用硬體(如 GPU 和 TPU)。

### 2.3 易於取得的先進技術

絕大多數處理資料和支援 AI 計算工作的系統都是開源軟體，包括 Spark、TensorFlow、MXNet、Caffe、PyTorch 和 BigDL。開源軟體讓所有組織和個人都可以使用最先進的技術，而不需要付出從頭開始的龐大開發成本，也沒有令人望而卻步的授權費用。

公共雲端服務(例如：AWS、Google Cloud 和 MS Azure)的廣泛可用性讓任何人都可以使用近乎無限的計算資源和存儲空間，而毋需建立大型資料中心。現在只要花費幾千美元，研究人員就可以隨時在多個 GPU 或 FPGA 上測試他們的演算法，這在十年前是無法想像的。

## 3 趨勢和挑戰

人工智慧已經開始應用在許多領域。我們期望未來的人工智慧能提供更廣泛的服務，從醫療保健到交通、從製造到國防、從娛樂到能源、從農業到零售。而由於近年來大型系統和 ML 框架又進一步推動了人工智慧的發展，因此我們也期望未來系統、安全、硬體架構能夠共同實現人工智慧的廣泛應用。不過，在那之前，我們需要解決以下趨勢所帶來的重大挑戰。

### 3.1 任務關鍵型(mission-critical)的 AI

AI 應用不斷在進步，從銀行業務到自動駕駛、從機器人輔助手術到家庭自動化，人工智慧未來可望推動更多跟人們(有時甚至跟人命)有關的關鍵任務應用。

隨著越來越多的人工智慧設置在動態的環境中，AI 系統需要在環境變化時不斷適應和學習新的「技能」，例如：自駕車可以即時學習其他成功處理狀況的自駕車，快速適應無法預期和危險的道路狀況(例如：車禍或路面出現漏油)。同樣地，智慧入侵檢測系統必須快速辨識並學習新的攻擊模式。而這些任務關鍵應用都必須處理吵雜的輸入並防禦惡意攻擊者。

**挑戰：**設計可做出及時、可靠和安全決策的 AI 系統，並且持續與動態環境互動進行學習。

### 3.2 客製化的 AI

從虛擬助理到自駕車，考慮使用者行為(例如：學習使用者口音的虛擬助理)和偏好(例如：學習使用者駕駛風格的自駕系統)的客製化決策越來越受重視。雖然這些客製化的系統和服務提供了嶄新的功能和顯著的經濟效益，但它們需要收集大量的敏感個資，而這些資料的誤用可能會為使用者的經濟狀況或心理健康帶來負面影響。

**挑戰：**設計不侵害使用者隱私和安全的客製化 AI 應用和服務系統。

### 3.3 跨組織的 AI

越來越多公司使用第三方資料來加強他們的智慧服務，例如：醫院分享資料以防止流行病爆發、金融機構分享資料以提高其詐欺檢測能力。這些應用的推廣將使資料孤島(自己收集資料、處理資料並提供服務)轉變成資料生態系統(使用不同組織擁有的資料進行學習並做出決策)。

**挑戰：**設計可以在不同組織的資料集上訓練而不侵害其機密的 AI 系統，並鼓勵可能互相是競爭對手的多個組織進行跨組織合作。

### 3.4 人們對 AI 的需求超越了摩爾定律

處理和存儲大量資料的能力是人工智慧最近成功的關鍵推動因素(詳見 2.1 節)，然而，由於以下兩種趨勢，要跟上資料的生成速度將變得越來越困難。

首先，資料量呈指數成長。2015 年 Cisco 的白皮書聲稱，物聯網設備產生的資料量會在 2018 年達到 400ZB，幾乎是 2015 年的 50 倍。根據最近的一項研究，到 2025 年，我們需要提升三到四個數量級的計算吞吐量來處理世界上所有基因組測序儀的總產出，意味著每年的計算資源至少要成長一倍。

其次，資料爆炸式增長的同時，過去快速進步的硬體技術卻開始停滯不前。DRAM 和硬碟的容量預計在未來十年內只會增加一倍，更糟的是，可能需要 20 年的時間才能讓 CPU 效能再增加一倍。硬體技術進步的減緩，讓存儲和處理所有資料的這件事開始變得不切實際。

**挑戰：**開發針對特定領域的結構和軟體系統，以滿足後摩爾定律時代 AI 應用的效能需求，包括用於 AI 計算工作的客製化積體電路、有效率地在邊緣處理資料的邊緣-雲端系

統、壓縮資料的技術。

## 4 研究機會

本節將從系統的角度討論：如何利用系統、安全性和架構方面的創新，來幫助解決前述的挑戰。如圖 1 所示，我們列出了九個挑戰(R1 到 R9)和三個研究主題(動態環境中的行為、安全的 AI、AI 專用的架構)，並描繪各個趨勢與挑戰和研究主題之間最常見的關係。

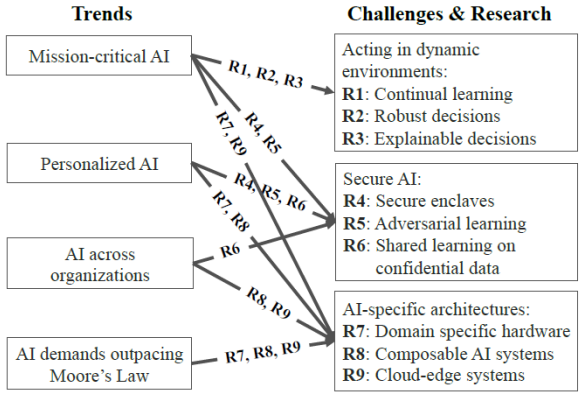


**Figure 1: A mapping from trends to challenges and research topics.**

### 4.1 在動態環境中運作

許多人工智慧的應用會在動態環境運作。動態環境即為經常迅速地發生不可預期變化的環境，且通常無法被重現。舉例來說，考慮多個共同為某辦公大樓提供保全服務的機器人，當有機器人損壞或增加新機器人時，其他機器人必須以協調的方式更新其導航、計劃和控制的策略，同樣地，當環境發生變化時，無論是由於機器人自身的行為抑或是外部狀況(例如：電梯停止運作或有惡意入侵者)，所有機器人都必須根據變化重新校準他們的行為。在這樣的動態環境中，即使是面對過去從未遇過的情況，AI 系統都必須能夠快速安全地做出反應來處理。

**R1：持續學習(終身學習)。**現今大多數的 AI 系統都是離線(offline)進行訓練，線上進行預測，例如：電影推薦、圖像辨識和語言翻譯等。也就是說，系統不會隨著資料的生成不斷學習，而是偶爾在截然不同且速度較慢的時間尺度進行學習。一般來說，模型通常一天或者一小時才更新一次，相對地，預測或決策卻以秒或亞秒的粒度發生。這使得它們不適合放在持續發生無法預期變化的環境，特別是任務關鍵型的應用。這些更具挑戰的環境(即動態環境)需要人工智慧不斷學習和適應非同步的變化。

部分在動態環境中學習的問題可以透過線上學習(online learning)來解決。當資料於某個時間點到達，線上學習可以一邊等待新資料，一邊更新模型。然而，傳統的線上學習並沒有特別去處理控制問題，像是 AI 的運作改變了環境(例如：機器人的行為所產生的變化)、或是決策結果會延遲的情況(例如：棋類比賽要到最後勝利或是失敗的時候才能針對某次的移動進行評估)。

這些更一般的情況可以用強化學習(RL)的框架解決。RL 的核心任務是學習一種「政策」—以最佳化最終的「回饋」(例如：避免碰撞或增加銷售額)為目標，將「觀察」(例如：行車紀錄器的輸入或使用者請求的內容)映射到一連串的「行動」(例如：減慢汽車速度或展示廣告)。RL 演算法透過考慮行為對環境的影響來更新政策，即使延遲也是如此，而如果是環境本身的變化導致回饋改變，RL 也會因此更新政策。RL 有著悠久的傳統和經典的成功故事，包括學習最優秀的人類玩家玩 backgammon、學習走路、以及學習基本運動技能等，然而，這些早期的設計需要針對不同的應用進行大幅度的調整。最近的研究開始結合了深度神經網路與 RL(即 Deep RL)，來開發可以適用於各種環境(例如：許多 Atari 遊戲)，甚至可以跨越不同應用領域(例如：(模擬)機器人的控制和機器人操縱技巧的學習)的訓練演算法，近期值得注意的成果當然少不了 Google 的 AlphaGo 擊敗圍棋世界冠軍，以及醫學診斷和資源管理的新應用。

不過，儘管取得了這些成功，目前尚未見到大規模的 RL 實際應用，造成這種情況的原因有很多，其中之一就是大型系統還沒有考慮這些案例。我們相信 RL 演算法在各方面不斷的進步和整合，加上系統設計的創新，可以促進 RL 的快速發展並推動新的 RL 應用。

RL 的系統。現今許多的 RL 應用仰賴模擬，為了在解決方案空間搜尋可能解法來「解決」複雜的任務，通常需要數百萬甚至上億次的模擬，例如：玩各種不同類型的遊戲、或是在機器人模擬器中測試不同的控制策略。這些模擬的持續時間可能小至幾毫秒，大至數分鐘(例如：要贏得一場遊戲可能需要數百個動作，卻只需要少數一些動作就可以輸掉它)。不過，現實世界中的 RL 系統終究必須在嚴格的時間限制內處理來自各種監控環境狀態的感測器的輸入，因此，我們需要能夠處理任意動態任務圖的系統，而且這些任務在時間、計算和資源方面的需求都是異構的。鑑於模擬的持續時間較短，我們應充分利用大型系統叢集來執行每秒數百萬次的模擬，然而，現有的系統都無法滿足這些要求，資料平行系統每秒處理的任務數量太少，而 HPC 和分佈式 DL 系統對異構和動態任務圖的支援有限，我們需要新的系統來滿足 RL 的效能需求。

模擬現實(SR)。與環境互動的能力是 RL 成功的基礎，不幸的是，現實世界的互動可能很慢(如數秒)或高風險(如不可逆的物理損害)，這些都與 RL 需要進行數百萬次互動以學習合理的策略相衝突。儘管已經存在減少學習策略所需互動次數的演算法，但普遍來說，我們更需要的是讓 AI 系統可以不斷地模擬和預測下一步行動結果的 SR 架構。

SR 不僅讓 AI 系統學習得更快，而且更安全。考慮一具清理環境的機器人，遇到了之前從未見過的物品，以新手機為例，機器人可以用手機進行物理實驗來確定如何抓住它，但這可能需要很長時間，而且可能會把手機摔壞。相較之下，機器人可以將手機的 3D 外觀掃描到模擬器中，然後做一些物理實驗來確定剛度、紋理和重量分佈，再使用 SR 學習如何成功地抓住它且不造成損壞。

重要的是，SR 與虛擬現實(VR)非常不同。VR 專注於模擬假想環境(例如：Minecraft)，有時結合現實世界的過去快照(例如：飛行模擬器)；SR 則專注於不斷模擬與 AI 應用互動的物理世界。SR 也與增強現實(AR)不同，後者主要將虛擬物件覆蓋到真實世界的圖像上。

SR 最大的系統挑戰是在不斷變化的真實環境中不斷調整模擬器參數，並在採取行動前執行無數次模擬。而且，當學習演算法與環境互動時，它會進一步獲得更多可用於改進模擬的知識，同時，由於 AI 應用的每個行動之間需要執行許多模擬，且每次會使用不同的計劃、對環境做出不同的「what-if」假設，因此，模擬必須比行動快很多。

**研究：**(1) 建立充分利用平行性、提供毫秒級延遲的動態任務圖、並於嚴格時限運行異構硬體的 RL 系統、(2) 建立能在持續發生不可預期變化的環境，依然忠實模擬且運行速度比行動快的 SR 系統。

**R2：穩健的決策。**隨著越來越多的 AI 應用替人們做出決策，它們需要對模糊的(或甚至是錯誤的)輸入和反饋保持穩健，當然，抗噪和穩健的學習本來就是統計學和機器學習的核心主題，不過我們可以透過增加系統支援，來大幅改善傳統的方法。我們可以建立跟蹤資料來源(provenance)的系統，以減少從資料來源映射到觀察結果所產生的不確定性，以及降低它們對狀態和回饋的影響，我們還可以追溯上下文資料(context)來提供資訊給針對特定來源的噪音模型(例如：遮擋(occluded)相機)。對於特定的 AI 系統，重要的兩個穩健性挑戰是：(1) 出現噪音或對立(adversarial)的反饋時穩健地進行學習、(2) 出現無法預料或對立的輸入時穩健地做出決策。

越來越多的學習系統使用從不可靠來源收集的資料，這些資料可能帶有不準確的標籤，有時候還會出現資料被人故意標錯的情況，例如：Microsoft 的聊天機器人 Tay 仰賴人機互動來開發豐富的自然對話功能，然而，當 Tay 暴露在 Twitter 的訊息底下後(意即用 Twitter 的訊息進行訓練)，便迅速呈現出另一種黑暗的個性(被 Twitter 的網友玩壞了)。

AI 系統除了要處理吵雜的反饋之外，另一項挑戰是處理未訓練過的輸入。常見的方法是，AI 系統先檢測請求輸入是否來自與訓練資料完全不同的分佈，是的話就採取安全措施(以自駕車為例，安全措施可以是減速或停止)，如果整個運作過程中有人類操作員，那麼一般會將決策系統的控制權讓給人類。明確地訓練模型(在發生意外時)拒絕做沒有信心的預測並直接採用預設的安全措施、建立將這些模型連接在一起的系統，既可以降低計算成本，又可以提供更準確和可靠的預測。

**研究：**(1) 將 AI 系統的結果變化(如回饋或狀態)與導致變化的資料連結起來，連結的資料越細越好(例如：資料 A 可分成資料 B 和資料 C，假使某結果 D 只與資料 B 有關，那就應將結果 D 和資料 B 連結起來，而非將結果 D 和資料 A 連結起來)，並使這個系統可以自主學習特定來源的因果噪音模型、(2) 設計開發系統時維持決策信賴區間以及標記不可預期輸入的 API 和語言。

**R3：可解釋(explainable)的決策。**除了做出黑箱的預測和決策之外，AI 系統還必須為它們的決策提供(對人類來說)有意義的解釋，這對於有大量管理需求的應用以及可能出現法律問題的保全系統或醫療保健等之應用尤為重要。這裡我們先區分 explainable 和 interpretable：後者也常常是被研究的對象，意思是 AI 演算法的輸出可以讓研究該題目的專家理解其所擷取的資料域的大致輪廓；而前者意味著可以用 AI 演算法的特定輸出去辨識其輸入的屬性，並且可以回答相反的狀況或假設性問題。以看 X 光片為例，我們可能希望得知器官的哪些特徵(例如：大小、顏色、位置、形狀)會導致特定診斷，以及如果這些特徵有細微的擾動，診斷將如何改變，我們可能也希望得知有沒有其他的狀況可能會有相同的結果，以及這些結果的相對合理性，此外，我們要考慮的通常不僅僅是 AI 系統為自身決策提供的解釋，還要考慮 AI 系統在這過程中可能額外負擔的資料。這裡我們討論了有關推論因果的研究方向，其對於許多未來的 AI 應用是不可或缺的，資料庫的診斷行為和資料來源的概念與之亦有直接連結。

可解釋決策的要素還有必須能夠「記錄」並「忠實地重播(replay)」導致特定決策的計算。這種系統可以針對過去的輸入重播任務(可以是有擾動的版本或甚至是相反的狀況)來辨識哪些輸入的特徵造成了哪些決定，以增進決策的可解釋性。以監視系統為例，為了辨識影像中造成錯誤警報的原因，可以將擾動引入影片以衰減警報訊號(例如：遮擋圖像的區域)或搜尋導致類似決策的相關歷史資料(例如：找出相關的輸入)。這種系統還可以改善統計診斷或新模型的訓練/測試，例如：設計適合(或不適合)解釋的模型。

**研究：**建立可以支援互動式診斷分析的 AI 系統，這個系統可以忠實地重播過去的執行、決策任務與擾動輸入來幫助找出導致特定決策的輸入的特徵。簡單來說，提供推論因果關係的系統支援。

## 4.2 安全的 AI

安全是個很大的主題，許多內容都會成為未來 AI 應用的核心議題，例如：關鍵任務的 AI 應用、客製化學習、跨多個組織的學習等，這些都需要具有強大安全性的系統。雖然有各式各樣的安全問題，不過我們在這裡主要只關注以下兩大類：第一類是破壞決策過程的完整性(integrity)，攻擊者可以透過破壞或控制 AI 系統、或改變系統的輸入來做到這一點，這樣系統就會在不知不覺中做出攻擊者想要的決定；第二類是攻擊者學習該 AI 系統訓練時所使用的機密資料、或學習其秘密模型。接下來，我們將討論防範這兩類攻擊的三個可能研究方向。

**R4：安全飛地(secure enclave)。**公共雲端的快速崛起和軟體堆疊複雜性的增長都使 AI 應用程式對攻擊的暴露大幅擴張。二十年前，AI 應用程式大多在一般的商業作業系統上(如 Windows 或 SunOS)執行，而且通常設置在組織後台的單一伺服器上。如今，組織在分佈式伺服器上的公共雲端執行

AI 應用程式，競爭對手可能與組織分享這些伺服器，而這些伺服器位於相當複雜的軟體堆疊上，作業系統本身可能也運行在虛擬機器管理程式(hypervisor)之上或容器(container)內。此外，這些 AI 應用程式直接或間接地使用了大量的其他系統，例如：擷取日誌、存儲和處理資料的平台，如果這些軟體的任何部分受到攻擊，那麼 AI 應用程式本身可能也會跟著損壞。

處理這種攻擊常用的方法是提供「安全飛地」—一種安全的執行環境—保護飛地內執行的程式不受飛地外的惡意程式碼影響。最近的例子是 Intel 的軟體保全擴充(SGX)，它提供了硬體隔離的執行環境，SGX 內部的程式碼可以安全地執行，即使是受損的作業系統或(在飛地外部執行的)虛擬機管理程式也無法查看這段程式碼或資料，SGX 還提供遠端認證，這是一種能使遠端使用者驗證飛地是否正在執行預期程式碼的協議。ARM 的 TrustZone 是硬體飛地的另一個例子。另一方面，雲端的銷售商開始使用一些可以物理保護的特殊元素，舉例來說，將程式碼或資料放置在安全的「保險庫」內，只有通過指紋或虹膜掃描認證的授權人員才能存取。

無論是何種飛地技術，開發人員必須信任在飛地內執行的所有軟體，事實上，即使在硬體飛地內執行，若執行的程式碼早已受到損害，也可能會因此洩漏機密資料或做出錯誤決策。由於較小的程式碼庫通常更容易保護，所以這裡的研究挑戰是利用加密技巧將 AI 系統的程式碼拆分為在飛地內執行的程式碼(越少越好)，以及在飛地外執行的程式碼。另一種確保飛地內的程式碼不會洩漏敏感訊息的方法是開發靜態與動態驗證工具以及沙盒。

請注意，除了最小化可信任的執行區段之外，還有兩個我們要分割程式碼的原因：第一、有些功能可能無法在飛地內使用，例如：執行深度學習(DL)演算法時處理 GPU 的程式、或者尚未經過審查或尚未移植到飛地的服務和應用；第二、雲端的銷售商所提供的安全版本可能會比常規版本貴很多。

**研究：**利用安全飛地確保資料的機密性、使用者隱私和決策的完整性，可能的方法為將 AI 系統的程式碼拆分成在飛地內執行的最小程式碼和飛地外執行的程式碼。

**R5：對立(adversarial)學習。**由於 ML 所使用的演算法會不斷調整參數，這樣的本質使 ML 系統容易遭受惡意改變訓練資料和決策輸入的攻擊，破壞其決策的完整性。這種攻擊大致可分成兩類：迴避(evasion)攻擊和資料毒化攻擊。

迴避攻擊發生在推理階段，攻擊者試圖製造會被學習系統錯誤分類的資料，以交通標誌為例，攻擊者可以稍微改變停止標誌的圖像，使得它對人類來說看起來仍是停止標誌，但是會被自駕車誤認成是讓行標誌。

資料毒化攻擊發生在訓練階段，攻擊者將有毒資料(例如：含錯誤標籤的資料)注入訓練資料集，導致學習系統學習錯誤的模型，攻擊者也因此擁有被 AI 學習系統錯誤分類的輸入資料。定期使用從不可靠或不可信的來源所收集的弱標記資料來重複訓練的學習系統(通常是為了處理非靜態輸入資料)，特別容易受到這種攻擊。由於新的 AI 系統會不斷與動態環境互動進行學習，處理資料毒化攻擊變得越來越重要。

直至今日，尚未出現抵禦迴避攻擊的可行方案，因此存在許多公開的研究挑戰：深入理解為何通常可以輕鬆找到對立樣本、調查有什麼方法或有什麼不同的方法組合可能可以有效地抵禦對立樣本、設計和開發系統方法來評估可能的防禦方式。而對於資料毒化攻擊，開放式挑戰包括：如何檢測中毒的輸入資料、以及如何建立一個能夠抵禦各種資料毒化攻擊的學習系統。更進一步，如果資料來源被認定是詐欺或出於管制原因而確定被收回，我們可以利用重播(詳見 R3：可解釋的決策)和增量計算來有效消除這些資料對學習模型的影響。如前所述，我們可以結合資料來源的建模和資料存儲系統的高效能計算來實現這種能力。

**研究：**建立在訓練和預測(如制定決策)期間可以穩健抵禦對立輸入的 AI 系統、設計新的機器學習模型和網路架構、追踪詐欺的資料來源，並在消除詐欺的資料後重播以重新制定決策。

**R6：用機密資料進行分享學習(shared learning)。**目前大部分的公司都是自己收集資料、進行分析，使用這些資料來實現新功能和產品，少數大型企業(例如：Google、臉書、微軟和亞馬遜)擁有為數龐大的資料，毋須跨組織合作便可發展人工智慧的應用，然而，並非所有組織皆是如此，因此我們期待未來有更多的組織共同收集珍貴的資料、更多的第三方資料服務、更多學習多個組織的資料(意即使用跨組織的資料進行訓練)所帶來的效益(詳見 3.3 節)。

透過與產業的互動，我們發現這種情形日益增加。有某家大型銀行提供了以下的情境：他們和其他銀行想要將他們的資料匯總在一起，使用分享學習來改善他們總體的詐欺檢測演算法。儘管這些銀行彼此是競爭對手，但這種「合作」能夠最大限度地減少詐欺行為，對於降低詐欺行為造成的損失至關重要。另外，有某家非常大的醫院也提供了類似的情境：互相競爭的多個醫院希望可以分享他們的資料，以訓練預測流感爆發的分享模型，使他們能夠改善對流行病的反應機制，舉例來說，迅速派遣疫苗接種巴士到重點區域來控制疫情。但與此同時，每家醫院都必須保護自己患者的機密。

分享學習的關鍵挑戰是如何在訓練過程中不洩露任何資料的相關訊息，來使用屬於不同組織(甚至是競爭對手)的資料學習模型。一種可能的解決方案是將所有資料匯集到某塊硬體飛地中然後學習模型，然而，由於硬體飛地尚未廣泛設置，因此這種解決方案並非總是可行的，而且有時候因為管理限制或是資料太過龐大，資料是無法移動的。

另一種可能的方法是使用安全的多方計算(MPC)。MPC 使得任何擁有資料的那一方都能計算包含他方資料的約定函數(joint function)，而無需學習他方的資料。不幸的是，雖然 MPC 對於簡單的計算還滿有效率的，但遇到複雜的計算(如模型訓練)，MPC 就會出現不小的成本。有趣的研究方向是如何將模型訓練劃分成「在本地端進行計算」和「使用 M

PC 進行計算」，來最小化 MPC 的複雜性。

儘管於訓練模型時不侵犯機密已經為實現分享學習跨出了重要的一步，不幸的是，要做的常常不只是如此：模型服務(模型所做出的推斷或決策)也可能洩漏資料的訊息。解決這個挑戰的其中一種方法是「差異隱私」，這是在統計資料庫的領域中一種熱門的技術，差異隱私有效地利用資料準確度來交換隱私性，為每個查詢添加噪音來保護資料隱私。差異隱私的核心概念是隱私預算(privacy budget)，以限制查詢次數的方式來確保隱私性。

為了將差異隱私應用於模型服務，有三個有趣的研究方向：第一、利用模型或預測原有的統計特性，在複雜的模型和推理上實作差異隱私；第二、建立工具和系統，讓實際應用可以輕鬆實現差異隱私，包括幫助應用挑選合適的隱私機制，還有自動將非差異隱私計算轉換為差異隱私計算，主要是因為雖然目前有許多差異隱私的理論研究，但實際應用的系統卻很少；第三、持續學習有個特點，資料隱私和時間可以是相依的，意即新資料的隱私遠比過去資料的隱私重要，例如：在股票市場或公開招標的場合，保護最新資料的隱私是極度重要的，相對地，歷史資料常常是被公佈的。因此可以利用這個特點，設計只適用於最新資料的決策之隱私預算，來開發新的差異隱私系統，或是進一步針對連續觀察和資料公開的特性，來深化差異隱私的概念。

即使我們能夠保護在訓練和決策過程的資料機密，這仍然不夠，現實中，由於競爭對手可能會從中受益，組織會抗拒分享他們的資料來改善模型。因此，我們除了保證的機密性，還要鼓勵組織分享他們的資料或資料的副產品，具體來說，我們要開發一些方法以確保組織能透過分享資料獲得(比不分享資料)更好的服務(即更好的決策)。首先檢測這個組織所提供資料的品質，這個問題可以用「留一驗證(leave-one-out)」來解決，無論這份資料是否包含在訓練集中，都要針對效能進行比較，接著提供用噪音破壞的決策，該決策被噪音破壞的程度與組織提供資料的品質成反比，以此來鼓勵組織提供更高品質的資料。這樣的獎勵措施必須放到設計機制的框架，使組織能夠研擬各自的資料分享策略。

**研究：**(1) 保證訓練或服務期間不會從任何資料來源洩漏資訊的跨資料來源學習 AI 系統、(2) 提供獎勵讓可能互相是競爭對手的多個組織分享資料或模型的 AI 系統。

## 4.3 為 AI 量身打造的架構

為了滿足人們對人工智慧的需求，我們必須在系統和硬體架構上進行創新，新的架構不僅要提高效能，還必須提供易於組合的豐富模組庫來簡化下一代 AI 應用的開發。

**R7：針對特定領域的硬體。**處理和存儲大量資料的能力是推動人工智慧近年來成功的關鍵因素(詳見 2.1 節)，然而，要跟上成長的資料量愈發困難，正如第 3 節所述，資料量呈指數級增長的同時，電腦產業 40 多年來效能-成本-能源的進步卻正逐漸邁向終點：

- 由於到達摩爾定律的終點，電晶體不會繼續變小
- 由於到達 Dennard 尺度定律的終點，功率限制了可以放在積體電路上的功能(單位能量可使用的功能有限)
- 積體電路從過去每片只能放一個低效能處理器到現在每片可以放十幾個高效能處理器。根據 Amdahl 定律，這個平行性終究會達到一個上限

因此，要快速提升處理效率，我們已不能再仰賴摩爾定律時代半導體製程的進步，必須透過電腦體系結構的創新來實現，而要繼續改善處理器的效能-能源-成本的途徑就只剩開發針對特定領域的處理器了。處理器只執行少數任務，但是可以把這些任務做得非常好，我們可以預期未來會有更多這種的異構處理器出現在伺服器上。針對特定領域處理器的濫觴是 Google 的 TPU，Google 於 2015 年開始將 TPU 設置在資料中心，提供給數十億人使用，TPU 執行深度神經網路推理的速度比同時期的 CPU 和 GPU 快 15 到 30 倍，且每瓦效能提高 30 到 80 倍。此外，微軟也宣佈他們的雲端系統 Azure 提供使用 FPGA 的服務。從 Intel 到 IBM，再到像 Cerebras 和 Graphcore 這樣的新創公司，許多公司都正為人工智慧開發專用的硬體。

有幾個正在開發的嶄新技術也希望能接替受到同樣限制的 DRAM：Intel 和美光共同開發的 3D XPoint 可以提供 DRAM 的 10 倍存儲容量且維持和 DRAM 差不多的效能；STT MRAM 的目標是接替快閃記憶體，不過它可能會有與 DRAM 類似的擴展限制。因此，雲端記憶體和儲存空間的階層結構(hierarchy)可能要有更多的層級，也可能要包含更多元化的技術，而鑒於這些處理器、記憶體和存儲設備的多樣性日益增加，要將服務映射到硬體資源會成為更具挑戰的議題。這些戲劇性的轉變都在在顯示一個訊息：與其使用標準的傳統框架(標準的傳統框架由交換器與其下的數十台伺服器所組成，每台伺服器配備兩個 CPU、一個 1 TB 的 DRAM 和一個 4 TB 的快閃磁碟機)，不如使用一些更靈活的功能元件(building block)來組成雲端計算系統。

舉例來說，UC Berkeley 的 Firebox 專案提出了一種連接數千個處理器、數千個 DRAM 和非揮發性儲存裝置的多框架超級電腦，並使用光纖提供低延遲、高頻寬和長物理距離，這樣的硬體系統可以根據各個系統軟體的計算服務，分配其相應類型和比例的特定領域處理器、DRAM 或 NVRAM。我們知道，AI 的計算工作通常需要龐大的記憶體(用空間換取時間)和五花八門的特定計算資源來維持效能，因此，把計算細緻分解並分配給相應的異構資源，可以有效地滿足日益多樣化的任務需求。

除了改善效能之外，新的硬體架構還要具備其他功能，像是安全性的支援。雖然 Intel 的 SGX 和 ARM 的 TrustZone 正在為硬體飛地鋪路，不過在它們完全被 AI 的應用程式接受之前，還有許多工作要做，特別是現有的飛地技術有各種資源限制，例如：可定址記憶體、還有它們僅適用於市面少數的 CPU。可能的研究方向有消除這些限制、提供統一的硬體飛地介面給各式各樣的專用處理器(包括 GPU 和 TPU)、開發新的安全功能(開源指令集處理器(如 RISC-V)可能會是開發安全功能的「遊樂場」)。

**研究：**(1) 設計強化 AI 應用的硬體結構，強化的目標可以是提高效能、降低功耗或增強安全性、(2) 設計利用「針對特定領域的硬體、細緻分解計算的架構、非揮發存儲技術」的 AI 軟體系統。

**R8：可組合的 AI 系統。**模組化(modularity)和可組合性(composition)在軟體系統的快速發展中扮演了關鍵的角色，它們讓開發人員得以用現有的元件快速建立或發展新的系統，例如：微核心作業系統(microkernel OS)、LAMP 堆疊、微服務架構(microservice architecture)和網際網路。相較之下，現今的 AI 系統是一體成形的，這使得它們難以開發、測試和發展。

有鑑於此，我們可以預期模組化和可組合性未來也會是提高人工智慧開發速度和使用率的關鍵，它們可以簡化人工智慧在複雜系統中的整合過程。以下我們將討論有關「模型的組合」以及「動作的組合」的研究問題。

模型的組合是邁向更靈活、更強大的 AI 系統的關鍵。將多個模型組合成一個模型服務系統，並提供不同形式的詢問(query)，這種設計讓我們得以在決策準確度、延遲和吞吐量三者間進行權衡，舉例來說，我們可以依序詢問這些模型，每個模型可能回傳準確度達到某個標準的決策，或是回傳「我不確定」，如果是後者，系統會將決策傳遞給下一個模型，我們把模型按照「我不確定」的比例(最高到最低)和延遲(最低到最高)進行排序，就能優化系統的延遲和準確度。

為了完全實現模型的組合，仍有許多挑戰需要解決，如：(1) 設計宣告式程式語言(declarative language)來抽取元件的拓撲、確認該應用的效能需求、(2) 準確地列出每個元件的效能模型，包括資源需求、延遲和吞吐量、(3) 調度元件間的執行排程、優化計算調度的演算法、將元件映射到可用資源，以達到延遲和吞吐量要求，同時盡量降低成本。

「動作的組合」是將一系列的基本決策或動作聚合成較大的基本單元，這樣的基本單元也稱為選項(options)。以行駛在高速公路的自駕車為例，動作可能有加速、減速、向左轉、向右轉、打左轉燈或打右轉燈等，那麼改變車道就可能是車子的一個選項(打燈加轉彎)；以機器人為例，動作可能有轉動或彎曲關節，那抓住物品就可能是機器人的一個選項。事實上，選項早已在階層學習的領域被廣泛研究，我們可以讓 AI 應用從現有的選項列表中選擇完成任務所需的選項，而不是從冗長的低階操作列表中選擇，從而大幅加快學習和適應新情境的速度。

Web 工程師開發應用程式時，只需短短數行程式碼就能調用功能強大的 Web API，我們也可以依樣畫葫蘆，編寫適當的選項、建立豐富的選項庫，來幫助未來 AI 應用的開發。此外，選項還可以提高回應的品質，因為在選項列表選擇下一個動作要比在原來的動作列表中選擇下一個動作簡單得多。

**研究：**(1) 設計模組化和能以靈活的方式組合模型和動作的 AI 系統、(2) 設計豐富的模型和選項庫，進而利用這些 API 大幅降低 AI 應用的開發成本。

**R9：雲端-邊緣系統。**語音辨識和語言翻譯等許多 AI 應用如今都設置在雲端，我們預期未來跨越邊緣設備和雲端的 AI 系統會迅速增加。最近，有越來越多雲端上的 AI 系統開始將部分功能轉移到邊緣設備，來提高隱私性、延遲性和安全性(包括失去網路時處理的能力)，例如：使用者推薦系統；另一方面，也有越來越多在邊緣的 AI 系統開始互相分享資料，並利用雲端的計算資源來更新模型和策略，例如：自駕車、無人機和家用機器人。

然而，開發雲端和雲端-邊緣系統非常困難，主要出於以下四種原因：首先，邊緣設備和資料中心伺服器的功能有很大的差異，我們預期這種差異只會逐漸加劇，因為手機和平板電腦等邊緣設備的功率或尺寸限制都遠比資料中心的伺服器嚴格得多；其次，邊緣設備的資源能力和軟體平台都極為多元，從驅動物聯網設備的超低功耗 ARM 或 RISC-V CPU 到自駕車中的強大 GPU，這種異構的特性使開發應用變得難上加難；第三，邊緣設備和資料中心的硬體和軟體更新週期相距過大(邊緣設備的更新週期顯然比較長)；第四，資料沒日沒夜地產生，存儲容量的提升卻逐漸減緩，存儲這些大數據的 cp 值越來越低。

要融合雲端和邊緣設備，有兩種解決方法。第一種方法是藉由可重新定向的軟體設計和編譯器技術來重新規劃邊緣設備的程式碼：(1) 軟體設計：為了處理邊緣設備異構的特性，以及更新正在邊緣設備上執行的應用程式，我們需要新的軟體堆疊，讓應用程式使用共同的 API 操作硬體功能，來(抽象地)移除設備的異構性；(2) 編譯器技術：開發可於邊緣設備運作時有效率地編譯複雜演算法的編譯器和即時(JIT)技術，可以利用一些現有的程式碼生成工具，例如：TensorFlow 的 XLA、Halide 和 Weld。

第二種方法是設計適合跨雲端和邊緣分區執行的 AI 系統。舉例來說，模型的組合(詳見 R8：可組合的 AI 系統)可以讓系統在邊緣執行較小但不太精確的模型，在雲端執行計算密集但更高精度的模型，這種架構可以降低做出決策的延遲，而不會影響準確度，目前已經有一些影像辨識系統開始使用這種架構。另一個例子是動作的組合(詳見 R8：可組合的 AI 系統)，讓系統在強大的雲端學習階層選項，然後在邊緣執行這些選項。

機器人是可以運用模組化雲端-邊緣架構的領域。目前開發機器人應用程式的開源平台很少，而 ROS 可以說是在這之中最熱門的平台，不過 ROS 僅限在本地端執行，且缺乏許多即時(real-time)應用程式所需的效能優化。為了結合人工智慧研究的新發展(例如：分享學習和持續學習)我們需要能夠跨越邊緣設備(例如機器人)和雲端的系統，讓開發人員可以在機器人和雲端之間無縫遷移功能，以優化決策延遲和學習的收斂時間，雲端可以使用機器人收集的訊息執行複雜的演算法來即時更新模型，而機器人可以繼續根據之前下載的策略在本地端執行動作。

為了處理邊緣設備所收集的大量資料，可以使用學習友善(learning-friendly)的壓縮方法來降低處理成本，例子有取樣(sampling)和繪製輪廓(sketching)(這兩個例子過去成功用

於分析工作負載)。研究方向是積極地利用取樣和繪製輪廓支援系統各種學習演算法和預測的情境，比較困難的部分是降低存儲成本，因為可能需要刪除資料，這裡的關鍵是有時我們無法得知未來這些資料會如何被使用，其實這本質上就是一個壓縮問題，只是必須根據 ML 演算法的目的進行壓縮。可以利用分佈式方法處理已經被取樣或繪製輪廓的資料，或使用加入特徵選擇或模型選擇協議的 ML 方法，來幫助解決這個問題。

**研究：**設計雲端-邊緣 AI 系統，(1) 利用邊緣設備降低延遲、提升安全性，並實作智慧資料保留(data retention)技術、(2) 利用雲端分享多個邊緣設備的資料和模型、訓練複雜的計算密集模型、採取高品質的決策。

## 5 結論

人工智慧在過去十年取得的驚人進展，讓人工智慧從實驗室裡的研究，成功轉變為毋需人力投入和監督的商業服務。而無論是 AI 系統、抑或是機器人，它們都沒有搶走人類的飯碗，事實上，它們提高了人類的績效、促進了新的合作形式。

要實現人工智慧成為我們生活得力助手的想像，必須克服無數艱鉅的挑戰，其中有許多挑戰與系統和基礎設施有關，這些挑戰是：AI 系統需要做出更快、更安全、更可解釋的決策；確保決策和學習的過程不受日益複雜的攻擊侵犯；面對摩爾定律的終結，持續提高計算能力；建立易於整合至現有應用的可組合雲端-邊緣系統。

本文分別就系統、架構和安全三個方面，提出了幾個具有潛力的開放研究方向來應對前述的挑戰。我們希望這些方向能激發新的研究成果、推動人工智慧的發展，使其更加強大、更好理解、更安全以及更可靠。