從AlphaGo看
人工智慧技術與應用

台大資訊管理系所
2018/5/18

I-Chen Wu
交通大學吳毅成教授

綱要

- 個人資料
- 回顧人工智慧
- 電腦遊戲與AI
- 深度學習 (Deep Learning)
- 深度強化式學習 (Deep Reinforcement Learning)
- 案例研究(Case Studies)
- 深度強化式學習應用類型
- 機會 (Opportunities)
- 挑戰 (Challenge)
- 結論

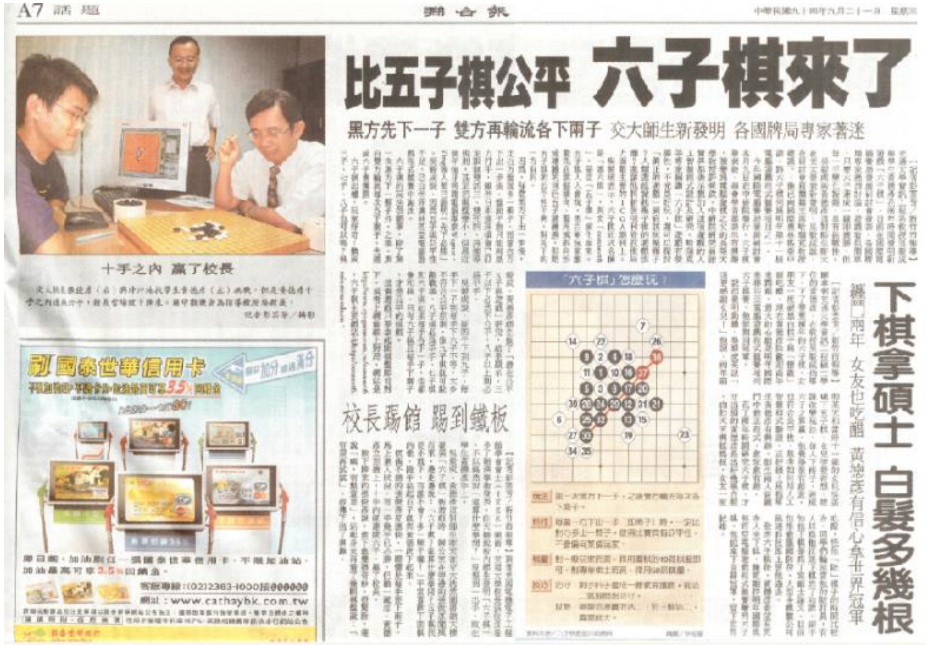
個人資料

- 學歷:
  - 台大電機學士(1982)
  - 台大資訊碩士(1984)
  - Carnegie Mellon University (CMU)電腦科學博士(1993)
- 現職:
  - 國立交通大學資訊工程系教授
  - 中華民國人工智慧學會理事長(2016-2017)
- 期刊編輯
  - Editor-in-Chief:
    - ICGA Journal (SCI).
  - Editorial Board
    - IEEE Transaction on Computational Intelligence and AI Games (SCI).
    - Journal of Experimental & Theoretical Artificial Intelligence (SCI).
    - Journal of Game Puzzle Design.

研究成果總結

- **發明六子棋**遊戲
- 多項遊戲在國際電腦對局遊戲競賽獲得冠軍 (**累計超過50冠軍**).
  - 2048程式:**全世界第一個打出65536磚塊!**
  - **電腦圍棋程式「CGI」**
    - 獲得**「世界智慧圍棋賽亞軍」**；**預賽全勝冠軍，擊敗騰訊公司的絕藝、DeepZenGo**
    - 第一個學界程式在正式的人機賽中，打敗職業九段棋士，以及**取得野狐圍棋網站「十段」**，並在該網站**多次擊敗世界排名前三名的棋士(柯潔、朴廷桓、半昱廷)**。
- 其他研究主題
  - 發展遊戲相關之P2P傳送系統、高速計算、雲端計算、行動軟體等系統
  - 發展機器學習相關應用問題，如工作排程、最佳監控涵蓋、機器手臂抓取
- 發表超過**120篇技術論文**, 其中超過**50篇SCI期刊論文**.
- 產學合作
  - 2016-目前: **中強光電、台積電、台達電、創義達科技、優必達、慧邦科技**
  - 2013-目前: 遊戲暨行動APP產業發展聯盟, 每年有5-10件產學合作計畫案 (≥NT$12,000,000)
  - 2013: 榮獲102年度科技部(原國科會)產學計畫「產學成果傑出獎」
  - 2012-2014: 交大／台達電整合型產學研究計畫總主持人 (≥ NT$25,000,000)
  - 2008-2011: 鈊象科技遊戲公司合作整合型產學計畫總主持人 (≥ NT$20,000,000)

六子棋



A7 話題　聯合報

比五子棋公平 六子棋來了

黑方先下一子 雙方再輪流各下兩子 交大師生新發明 各國牌局專家著迷

十手之內 贏了校長

校長踢館 踢到鐵板

下棋拿碩士 白髮多幾根

吳□兩年 女友也吃醋 黃煒彥有信心拿世界冠軍

「六子棋」怎麼玩？

7
14
22
26
9
2
4
18
36
11
1
10
16
37
5
3
8
17
20
30
28
24
29
12
31
21
6
25
33
13
15
27
32
19
23
34
35

奧林匹亞電腦賽局 交大5金2銀

聯合報 B 新竹一運動

【記者李青霖／新竹報導】交通大學資訊工程系教授吳毅成帶領學生團隊，到日本橫濱慶應義塾大學參加國際奧林匹亞電腦賽局競賽，拿到5金2銀獎牌，吳教授會中發表的論文，也獲最佳論文獎。

「這是歷來最佳成績」吳教授說，拿金牌項目包括：六子棋、禁圍棋（NoGo）、Nonogram、暗棋和麻將；銀牌部分是：禁圍棋（另一組）和象棋。

吳毅成是六子棋發明人，2005年發表後，發展出六子棋程式：「交大六號」，曾獲2006、2008年奧林匹亞電腦賽局冠軍，今年再奪冠。

他說，去年利用國科會產學合作計畫，發展手機、平板電腦六子棋程式，將「交大六號」改寫、濃縮到平板，這樣的改變，讓計算速度慢了近10倍，加上記憶體不足，賽前一周仍考慮是否參賽？「想不到還可以跑第一」，驗證未來推廣到手機、平板電腦上仍具潛力。

吳教授說，禁圍棋（NoGo）、Nonogram、暗棋、麻將項目，交大首次組隊參賽，都拿冠軍：暗棋是東方棋類遊戲，今年吸引來自法國等10支隊伍參賽，博士生曾汶傑所寫的程式「DarkNight」沒有敗場。

他說，團隊能拿到佳績，除了深入了解各種技術，最重要「武器」是由學生劉浩雲、康皓華、廖挺富等人發展出來的「通用型的遊戲軟體發展平台」，可讓遊戲發展者專注在遊戲的人工智慧技術上，且簡化軟體處理及除錯工作。

交大教授吳毅成（左起）、魏廷翰（行動六號作者之一），與大會主席海瑞克教授合影。圖／交大提供

交大吳毅成教授 奧林匹亞賽揚威

6子棋程式 我國際奪金

潘國正／竹市報導

交通大學資訊工程系吳毅成教授的6子棋人工智慧程式：交大六號(NCTU6)，在義大利杜林舉辦的第11屆奧林匹亞電腦競賽(11th Computer Olympiad)，為台灣獲得唯一的1面金牌。這項競賽包括電腦西洋棋、象棋、圍棋等。

吳毅成教授表示，他研究6子棋程式在這次世界性競賽的致勝原因，是採用雙迫著(double threat)攻擊法：這也就是說每次下出2子迫使對方2個子必須全部用來阻擋，然後藉由連續雙迫著壓迫對手來贏得勝利。

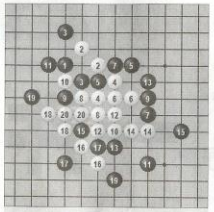
打破5子棋「先下手為強」的定律，吳教授和女兒下棋體悟出來的6子棋遊戲，去年九月發表世界第一篇介紹6子棋論文後，引起相關學者的高度興趣。

這個新遊戲的玩法非常簡單，第一手黑方只下1子，接下來雙方輪流各下2子。這玩法明顯提高遊戲的公平性，由於1次下2子，遊戲複雜度非常高。

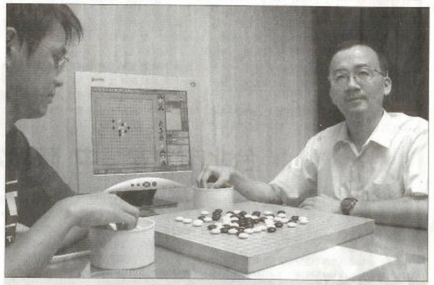
因此，發表後不久就被國際認可，並列入奧林匹亞電腦競賽的項目之一。也被登入線上維科百科全書，這可能是由台灣自行發展出來，並經國際認可的棋類遊戲。

吳教授發表6子棋後，發展十分快速。除了台灣成立聯誼會及六子棋論壇網站(http://groups.msn.com/connect6)，大陸也成立6子棋論壇網站(http://post.baidu.com/f?kw=%C1%F9%D7%D3%C6%E5)。

在遊戲網站方面，國內群想遊戲網站(cycgame.com)提供國內玩家線上玩6子棋。



▲交大吳毅成教授奪得義大利6子棋競賽金面，獲勝的原因是採「雙迫法」，每次下出2個迫使對方2個子必須全部用來阻擋；然後藉由連續雙迫著壓迫對手來贏得勝利。例如圖中白從第14手棋後，全為雙迫著黑不得不應。（吳毅成提供）



▲交大教授吳毅成（右）發明6子棋新遊戲，打破5子棋「先下手為強」的慣例，這次在義大利舉辦第11屆奧林匹亞電腦競賽獲得台灣唯一的金牌。（陳愛珠攝）

連成6子才算贏 更公平

陳愛珠、潘國正／竹市報導

5子棋是許多人的童年棋藝經驗，但常被人詬病的是公平性不夠，先下者(通常是持黑子者)較有利。交大資工所教授吳毅成和女兒下5子棋時，女兒提議每人下2子，連成6子才算贏，這個創意讓他研究出變化多端的6子棋。

吳毅成研發的6子棋，由黑方下1子，之後黑白雙方輪流每次各下2子，連成6子者獲勝，除了比5子棋公平外，遊戲法高達2萬5千多種，變化性和挑戰性更大。

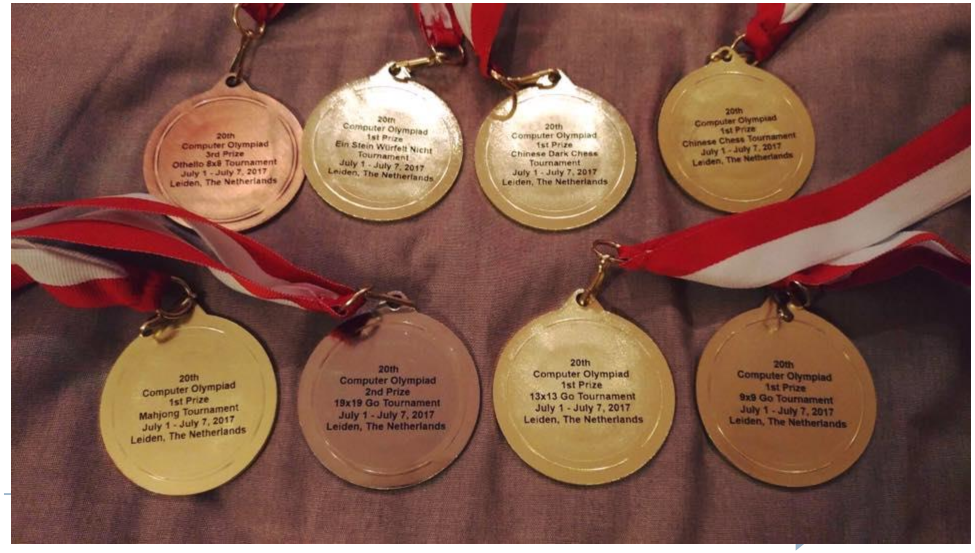
回想3年前，他和女兒下5子棋，女兒異想天開地建議：「每個人下2子連成6個才贏，好嗎？」吳毅成心想，5子棋先下1子已是不太公平的事，一次下2個，豈不更不公平。

他同時想到，若第一手黑方只下1子，接下來才輪流各下2子，類似網球搶7的開球方一樣，這是不是就公平了？試著下了幾次後，對其中的公平、複雜，充滿好奇與興趣，決定弄清楚而研究有成。

如何驗證或論證6子棋是一個公平且複雜的棋賽？他設計一個人工智慧程式來玩這遊戲。九十三年初，他的碩士班學生黃德彥參與這項研究，並作為個人碩士論文。

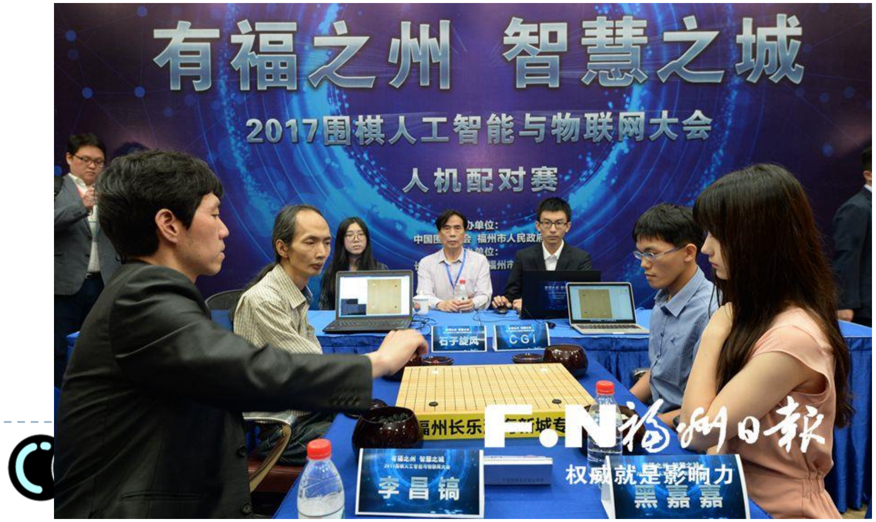
去年年初，他們完成第一個6子棋程式，一邊與電腦對弈，一邊修正該程式。吳毅成並設計出1283種開局樣式，讓電腦對電腦下。直到目前為止，還沒發現，對先下或者是後下哪一方特別有利。

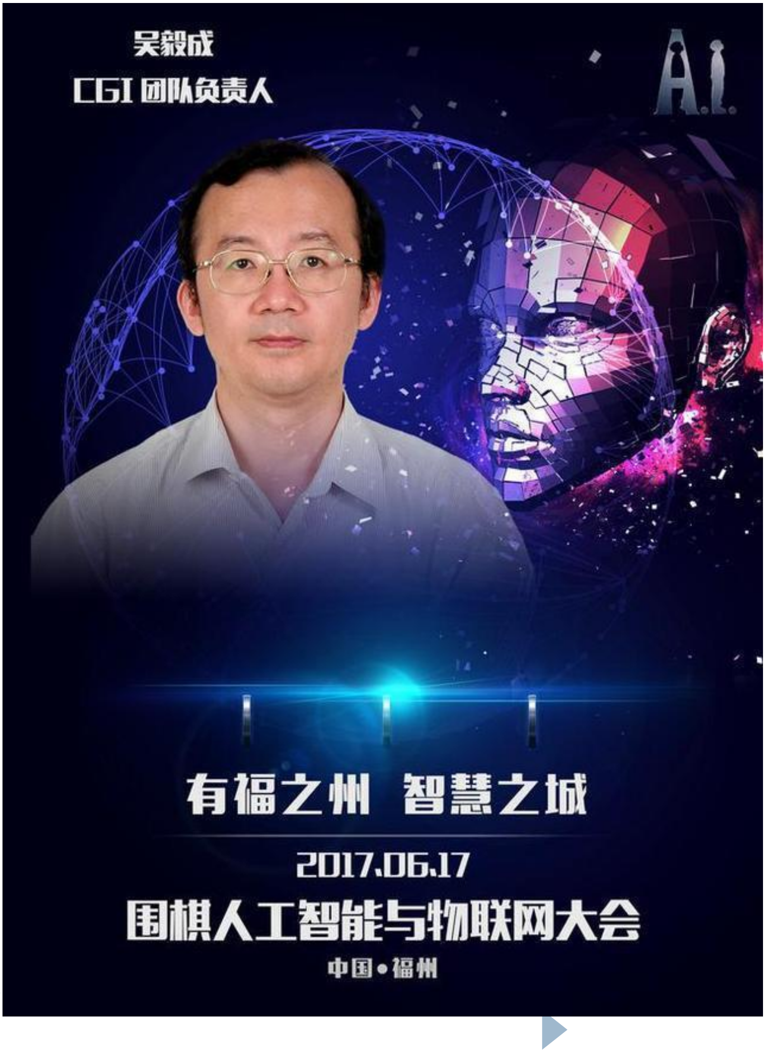
去年九月，吳毅成將六子棋新遊戲法發表在第十一屆國際電腦賽局發展(Advances in Computer Games)研討會，主席哈瑞克教授(Prof. Herik)與吳毅成討論後，將新創遊戲發明刊登在賽局領域最主要的ICGA期刊上，並申請為第十一屆奧林匹亞電腦遊戲程式競賽項目之一。


20th Computer Olympiad 3rd Prize Othello 8x8 Tournament July 1 - July 7, 2017 Leiden, The Netherlands
20th Computer Olympiad 1st Prize Ein Stein Würfelt Nicht Tournament July 1 - July 7, 2017 Leiden, The Netherlands
20th Computer Olympiad 1st Prize Chinese Dark Chess Tournament July 1 - July 7, 2017 Leiden, The Netherlands
20th Computer Olympiad 1st Prize Chinese Chess Tournament July 1 - July 7, 2017 Leiden, The Netherlands
20th Computer Olympiad 1st Prize Mahjong Tournament July 1 - July 7, 2017 Leiden, The Netherlands
20th Computer Olympiad 2nd Prize 19x19 Go Tournament July 1 - July 7, 2017 Leiden, The Netherlands
20th Computer Olympiad 1st Prize 13x13 Go Tournament July 1 - July 7, 2017 Leiden, The Netherlands
20th Computer Olympiad 1st Prize 9x9 Go Tournament July 1 - July 7, 2017 Leiden, The Netherlands

中韓台人機配對賽

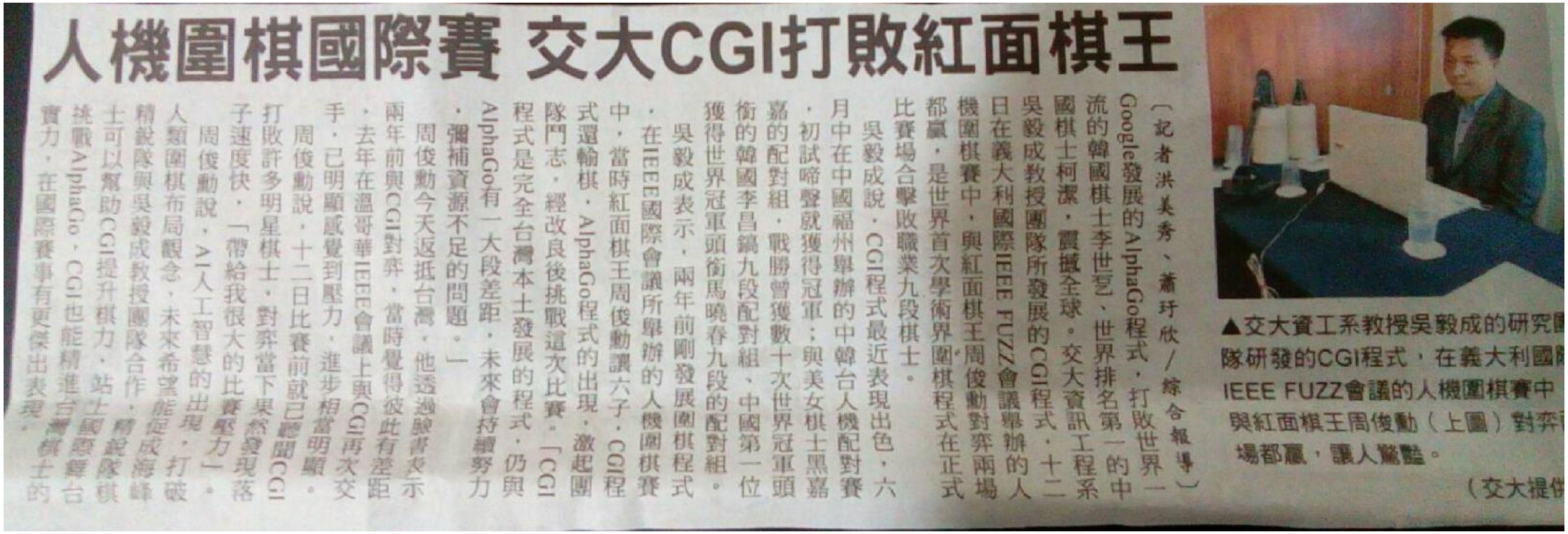
棋人工智能与
弈城围棋网
eweiqi.com


有福之州 智慧之城
2017围棋人工智能与物联网大会
人机配对赛
石子旋风
CGI
李昌镐
黑嘉嘉
F.N 福州日報
权威就是影响力


吴毅成
CGI 团队负责人
A.I.
有福之州 智慧之城
2017.06.17
围棋人工智能与物联网大会
中国•福州

IEEE FUZZ 人機圍棋賽

- 這是全世界第一次學界圍棋程式在正式比賽的場合中，擊敗職業九段棋士。



人機圍棋國際賽 交大CGI打敗紅面棋王

（記者洪美秀、蕭玗欣／綜合報導）Google發展的AlphaGo程式，打敗世界一流的韓國棋士李世乭，世界排名第一的中國棋士柯潔，震撼全球。交大資訊工程系吳毅成教授團隊所發展的CGI程式，十二日在義大利國際IEEE FUZZ會議舉辦的人機圍棋賽中，與紅面棋王周俊勳對弈兩場都贏，是世界首次學術界圍棋程式在正式比賽場合擊敗職業九段棋士。

吳毅成說，CGI程式最近表現出色，六月中在中國福州舉辦的中韓台人機配對賽，初試啼聲就獲得冠軍；與美女棋士黑嘉嘉的配對組，戰勝曾獲數十次世界冠軍頭銜的韓國李昌鎬九段配對組、中國第一位獲得世界冠軍頭銜馬曉春九段的配對組。

吳毅成表示，兩年前剛發展圍棋程式，在IEEE國際會議所舉辦的人機圍棋賽中，當時紅面棋王周俊勳讓六子，CGI程式還輸棋，AlphaGo程式的出現，激起團隊鬥志，經改良後挑戰這次比賽。「CGI程式是完全台灣本土發展的程式，仍與AlphaGo有一大段差距，未來會持續努力，彌補資源不足的問題。」

周俊勳今天返抵台灣，他透過臉書表示，兩年前與CGI對弈，當時覺得彼此有差距，去年在溫哥華IEEE會議上與CGI再次交手，已明顯感覺到壓力，進步相當明顯。

周俊勳說，十二日比賽前就已聽聞CGI打敗許多明星棋士，對弈當下果然發現落子速度快，「帶給我很大的比賽壓力」。

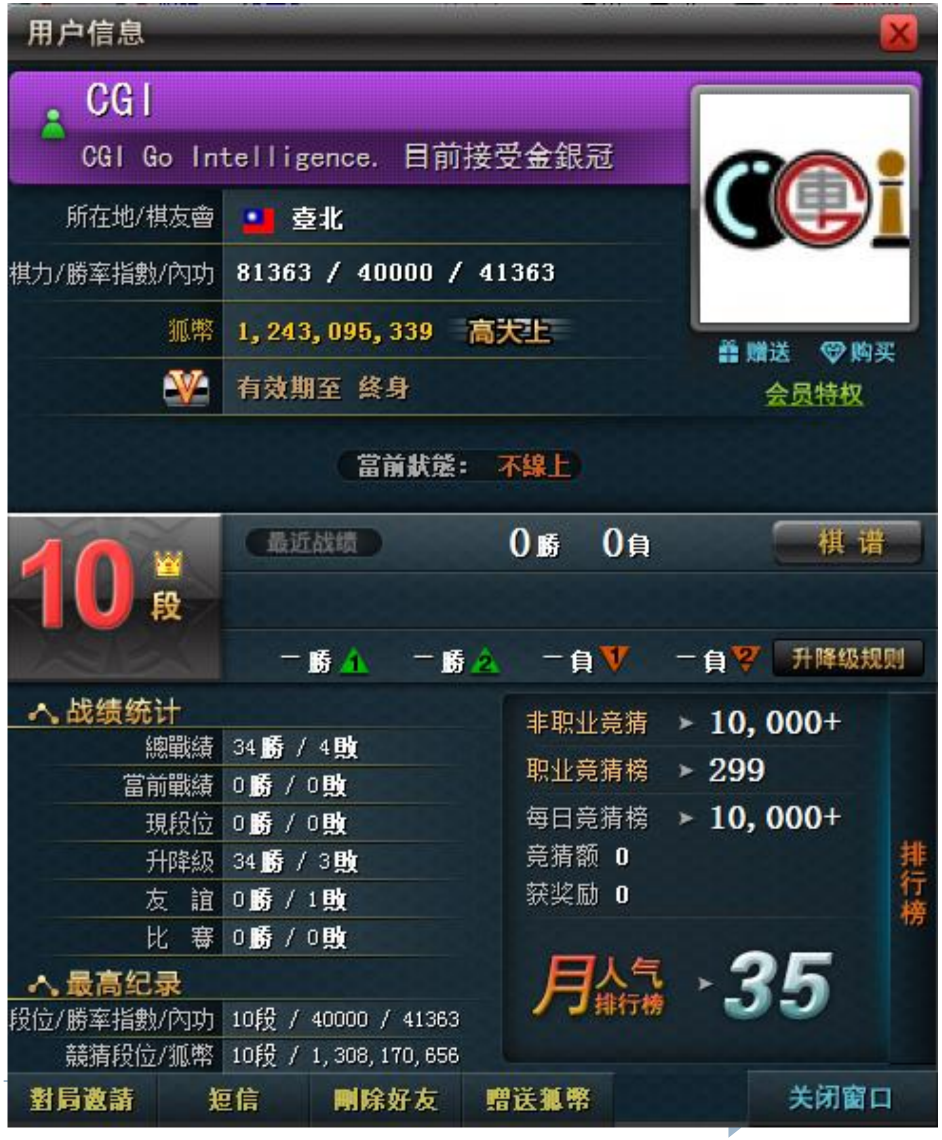
周俊勳說，AI人工智慧的出現，打破人類圍棋布局觀念，未來希望能促成海峰精銳隊與吳毅成教授團隊合作，精銳隊棋士可以幫助CGI提升棋力，站上國際舞台挑戰AlphaGo，CGI也能精進台灣棋士的實力，在國際賽事有更傑出表現。

▲交大資工系教授吳毅成的研究團隊研發的CGI程式，在義大利國際IEEE FUZZ會議的人機圍棋賽中與紅面棋王周俊勳（上圖）對弈場都贏，讓人驚豔。

（交大提供

世界智能圍棋賽亞軍

獲野狐圍棋網站十段頭銜



CGI

CGI Go Intelligence. 目前接受金銀冠

所在地/棋友會 臺北

棋力/勝率指數/內功 81363 / 40000 / 41363

狐幣 1,243,095,339 高大上

有效期至 终身

當前狀態: 不線上

10 段

最近战绩 0 勝 0 負

一勝 1 一勝 2 一負 1 一負 2

战绩统计

總戰績	34 勝 / 4 敗
當前戰績	0 勝 / 0 敗
現段位	0 勝 / 0 敗
升降級	34 勝 / 3 敗
友 誼	0 勝 / 1 敗
比 賽	0 勝 / 0 敗

最高纪录

段位/勝率指數/內功	10段 / 40000 / 41363
競猜段位/狐幣	10段 / 1,308,170,656

非职业竞猜	10,000+
职业竞猜榜	299
每日竞猜榜	10,000+
竞猜额	0
获奖励	0

月人气排行榜 35

- 多次戰勝棋手: (gorating.org, 2017/10)
  - 柯潔(世界排行第一名)
  - 樸廷桓(世界排行第二名)
  - 芈昱廷(世界排行第三名)

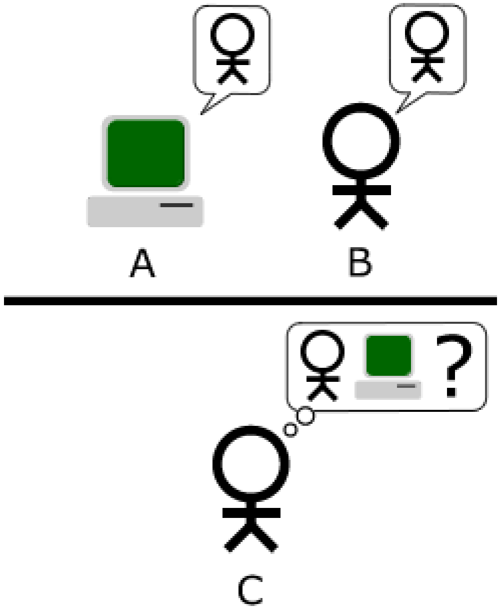
回顧
人工智慧

人工智慧定義

- John McCarthy 教授 (人工智慧之父) :
  - 具有智慧的機器(或具智慧的電腦程式)之科學與工程
  - The science and engineering of making intelligent machines, especially intelligent computer programs.
- Russell and Norvig (Artificial Intelligence: A Modern Approach):
  - 人性化的思維、理性化的思維
    - Thinking Humanly, Thinking Rationally
  - 人性化的行為、理性化的行為
    - Acting Humanly, Acting Rationally
- 具體而言，研究類似人類智慧思維/行為的機器(電腦程式)，李開複的定義：
  - 感知 (視覺、語音、語言)
  - 決策 (識別、推薦、預測；如人臉辨識、下棋、股市預測)
  - 回饋 (學習、邏輯推論、機器人、自動化)

Turing Test (杜林測試)

- 如何定義人工智慧? How to define “artificial intelligence"?
  - 若一個機器有智慧, 應該讓人無法分別它與人。
  - If a machine is intelligent, it cannot be distinguished from a human.
- "Can machines think?" – Turing’s question in 1950.
  - But, hard to define. So, proposed the Turing test as follows.
    - A human judge engages in a natural language conversation with one human and one machine,
      - each of which tries to appear human.
    - All participants are placed in isolated locations.
    - If the judge cannot reliably tell the machine from the human, the machine passes the test.
    - The conversation is limited to a text-only channel
  - Computer arguably passes Turing Test for the first time, 2014.
- "Are there imaginable digital computers which would pass in the Turing test" – Turing's new question.



人工智慧的應用

- Game Artificial Intelligence
- Strategic Planning
- Speech/Pattern Recognition, Computer Vision, Virtual Reality and Image Processing
- Optimization Problems
- Natural Language Processing, Translation and Chatterbots
- Expert System
- Artificial Creativity
- Factory Planning and Scheduling
- Applied to other applications:
  - Drive Automation, Drone, Robotics
  - Medical diagnosis
  - Big Data, IoT
  - FinTech
  - Email spam filtering

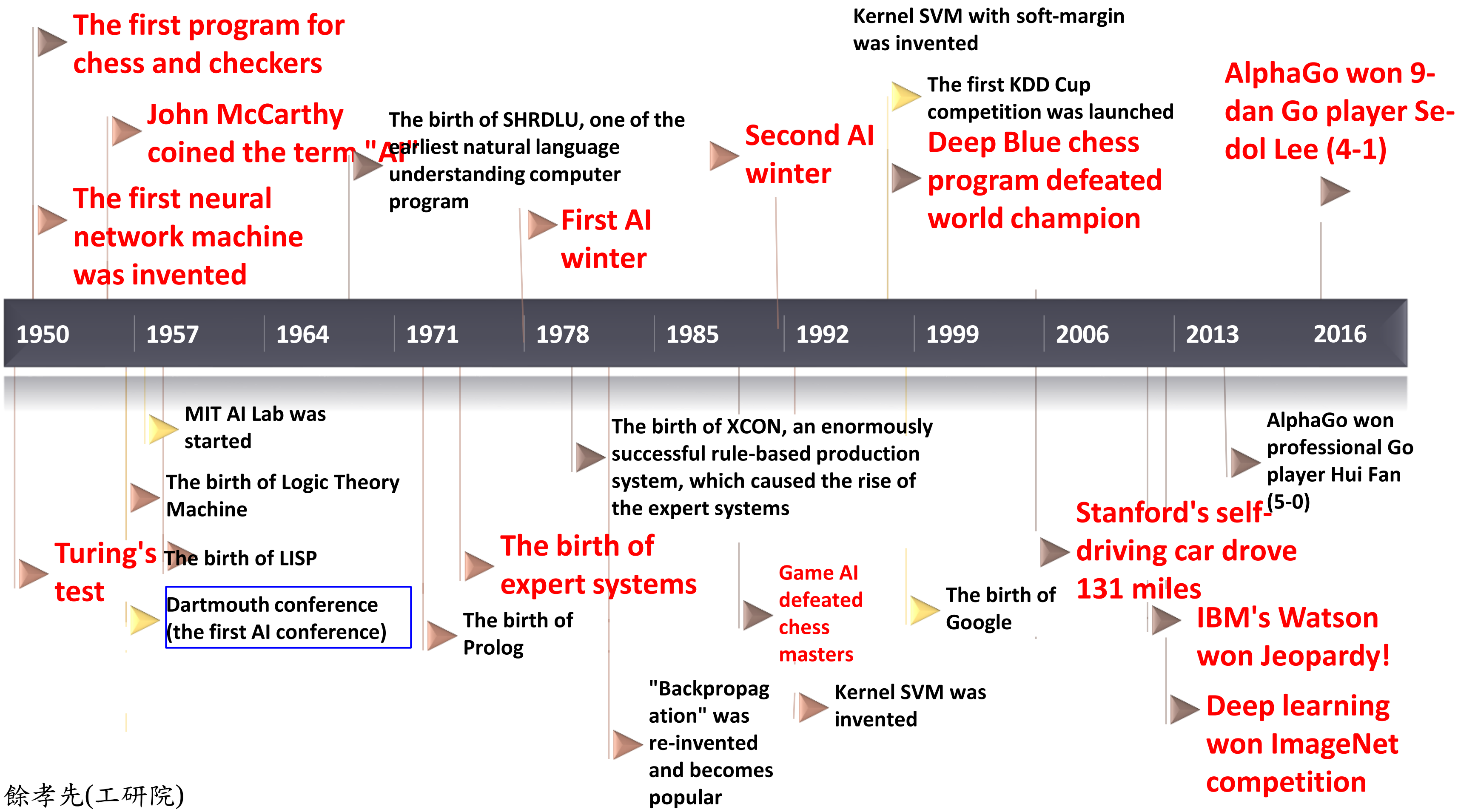
核心技術 (1/2)

- Planning
  - Deduction, reasoning, problem solving
  - Alpha-beta Search, Proof-Number Search
  - Monte-Carlo Tree Search
- Machine Learning
  - Supervised Learning
    - Support Vector Machine, Linear/Quadratic Regression
  - Unsupervised Learning
    - Clustering
  - Reinforcement Learning
    - Temporal Difference Learning, Monte-Carlo Tree Search
- Neural Network (including Deep Neural Network)
  - (described in more detail later)

核心技術 (2/2)

- Mathematical Optimization
  - Genetic Algorithm, Immune Algorithm
  - Fuzzy Set
  - Constraint Satisfactory Problem (CSP)
  - Scheduling, Particle Swarm Optimization (PSO)
- Pattern/Speech Recognition
  - Hidden Markov Model
  - Bayesian Learning
- Natural language processing
  - Decision Tree
  - Neural Network (LSTM, RNN etc)
- Knowledge representation
  - Expert System, Ontology
- Data Mining
  - Big Data, Social intelligence

人工智慧發展歷程 (~2016)



1950 1957 1964 1971 1978 1985 1992 1999 2006 2013 2016

The first program for chess and checkers

John McCarthy coined the term "AI"

The first neural network machine was invented

The birth of SHRDLU, one of the earliest natural language understanding computer program

First AI winter

Second AI winter

Kernel SVM with soft-margin was invented

The first KDD Cup competition was launched

Deep Blue chess program defeated world champion

AlphaGo won 9-dan Go player Se-dol Lee (4-1)

Turing's test

MIT AI Lab was started

The birth of Logic Theory Machine

The birth of LISP

Dartmouth conference (the first AI conference)

The birth of Prolog

The birth of expert systems

The birth of XCON, an enormously successful rule-based production system, which caused the rise of the expert systems

"Backpropagation" was re-invented and becomes popular

Game AI defeated chess masters

Kernel SVM was invented

The birth of Google

Stanford's self-driving car drove 131 miles

IBM's Watson won Jeopardy!

Deep learning won ImageNet competition

AlphaGo won professional Go player Hui Fan (5-0)

修改自: 餘孝先(工研院)

電腦遊戲與AI

深度學習 ▶

Schaeffer & Herik [2002]:

“Chess is to AI as drosophila (the fruit fly) is to genetics”

“西洋棋之於AI,相當於果蠅之于基因”

“西洋棋是AI的果蠅”

更精確的說:

“圍棋才是AI的果蠅”

“Computer Games是AI的果蠅”

AI的兩個重要里程碑

1. 1997年:
   - IBM深藍(Deep Blue)擊敗西洋棋棋王Kasparov
2. 2016年:
   - Google Deepmind’s AlphaGo擊敗李世石
     - 這十年獲得最多圍棋冠軍頭銜

電腦人腦棋賽・機器贏了頭盤

世界棋王下得辛苦・國際網路「實況轉播」

【本報記者李勇費城報導】國際商業機器公司（IBM）的「深藍」西洋棋計劃挑戰棋王蓋瑞・柯斯巴魯夫（Garry Kasparov）的比賽，首場IBM「深藍」旗開得勝。

棋王柯斯巴魯夫是在下棋三小時後，知道大勢已去，雖然整個賽程有七個小時，但他決定放棄比賽，承認失敗。

這場電腦與人腦棋賽於十日下午三時準時在費城市會議中心展開，操縱電腦的是「深藍」計劃五人小組成員許峰雄，他與棋王對坐在棋桌兩端，開賽前五分鐘，兩人迅速走動，容許新聞記者從不同角度拍攝兩人的照片，五分鐘後棋賽進入情況，需要思考才可移動棋子，站在旁邊的裁判立即命記者向後退回記者席，不讓他們照相以免擾亂棋「王」的思考。

IBM「華生研究中心」數十名研究員在「深藍」計劃負責人譚崇仁安排下，全部出動配合支援並接待三百多名購票入場的電腦專家及西洋棋愛好者。讓他們在另一大廳面對電視螢光幕把棋賽進行看得一清二楚。

觀看棋賽的大廳與比賽場地完全隔離，IBM特別請了美國棋王耶舍・斯來維及一名黑人西洋棋高手向觀眾解釋戰況，說明棋政、分析優劣。預測勝負。

另外一個小組研究人員在另一個房間把比賽現場的實況透過國際電腦網路立即傳送到世界各地，讓關心這場賽事的人，不論在任何地方，都可立即從網路上知道戰況與結果。

棋賽由IBM「深藍」先下，華裔研究員許峰雄在電腦鍵盤上按了幾下，按照電腦的指示先走第一步，然後以筆記下來，棋王立即回應第二步棋，又以筆記下來，棋賽開始廿分鐘後，「棋王」開始進入長考，不久便離座到後面休息十分鐘後才出來走下一步棋，電腦的反應很快，跟著又下一步，於是棋王又開始長考。

根據耶舍・斯來維的講解，電腦在開始一個小時的攻擊凌厲，因此棋王應戰得十分辛苦，他之所以離席休息，目的是想出一個可以突破電腦系統的方法。果然，他休息出來後，連續下了幾步狠棋，直搗對方陣地，顯示出棋王不凡的思考與傑出的棋藝。但是電腦快速冷靜的技巧明顯超過棋王，經過三小時的鏖戰之後，棋局上雖看不出誰有敗象，但棋王心裡有數，宣布投降。

「深藍」計劃的經理譚崇仁在賽後興奮地說：他們只要贏了第一場與第二場的比賽，整個賽局就可控制下來，他仍然估計，這次六場比賽，IBM會以四比二的贏率擊敗棋王，為電腦分析寫下新的一頁。

世界西洋棋冠軍柯斯巴魯夫（左）十日不敵電腦，他在費城舉行的與美國電腦計算協會的國際電腦西洋棋冠軍IBM的「深藍」系統對弈六局中，第一局提前認輸。圖爲比賽的實況，代表「深藍」系統在棋盤上移動棋子者是該系統的設計人之一許峰雄。（美聯社）

對弈第六回合　電腦妙算十九步定江山

深藍挑戰成功

棋王輸在動了情緒

由來自港、台華裔電腦專家等所領導的I「深藍」電腦經過六局鏖戰後擊敗世界西王卡斯帕洛夫，創下電腦首度打敗人類棋歷史性紀錄，並造成卡斯帕洛夫稱霸西洋以來首次嘗到敗績。

「深藍」十一日下午只花了一小時就擊斯帕洛夫，以六局總積分三點五比二點五績，迫使卡斯帕洛夫稱臣。第六局最後一鍵決賽進行時，現場及透過電視螢幕前的棋迷屏息觀戰，棋賽在第十九手分出勝負王棄子投降。

卡斯帕洛夫認輸後表情懊惱、震驚，這位去年曾以四比二戰績在費城擊敗「深藍」並誇言「廿世紀結束前不可能有人或電腦擊敗他」的棋王，隨後在記者會中說：「我必須向棋迷致歉，我對最後這一局的表現感到慚愧。」

卡斯帕洛夫解釋說，他是人，當他看到有些情況超越他所理解的範圍時，他會感到害怕。他又說，這次棋賽可以說是「世界上棋藝最高的人在壓力之下俯首稱臣。」

專家們認為卡斯帕洛夫是在生理和心理的雙重壓力下，輸掉第六局比賽。在場觀棋的法國西洋棋高手勞特爾說，縱橫棋壇十餘載的棋王在下第十九步棋就俯首稱臣，令人震驚，因為這局似乎沒有輸，過去都是這麼下，這是眾所週知的棋步。勞特爾說，卡斯帕洛夫有一點失常，他一直想調整步伐適應電腦但不夠積極。

在這局持黑子後攻的卡斯帕洛夫原擬迂迴佈陣，但開賽後不久即犯下失誤，讓深藍在第八步棋犧牲騎士換取卡斯帕洛夫的卒子，打亂了佈局，從佈陣轉為交戰狀態。卡斯帕洛夫陸續取下深藍的城堡、主教和騎士，卻在第十八步棋輸掉皇后，但還不至於輸陣。然而，深藍第十九步棋即把卡斯帕洛夫的國王困坐愁城，卡斯帕洛夫被迫棄子稱降。

「深藍」計畫主持人譚崇仁在賽後記者會上表示，「深藍」計畫全體人員對能夠贏得這次比賽勝利，覺得十分驕傲，並且感到榮幸能參與這一歷史事件。譚崇仁對於卡斯帕洛夫這六場的表現給予極高度評價，強調棋王智力過人，了解電腦將可以帶人類發展到什麼起步。

卡斯帕洛夫也抱怨未能取得深藍過去比賽時的棋譜。根據西洋棋比賽規則，雙方可以要求研究對方過去比賽的棋路。卡斯帕洛夫說，他最大的錯誤就是沒有提出條件，讓這場比賽進行更公平些。卡斯帕洛夫不排除再與深藍對弈的可能，但他堅持在不同的條件下對決，例如讓他和電腦先作練習賽。

勝方「深藍」計畫小組將可獲七十萬美元的獎金，IBM公司已決定將這筆獎金用於電腦研究，至於卡斯帕洛夫則可獲四十萬美元的獎金。

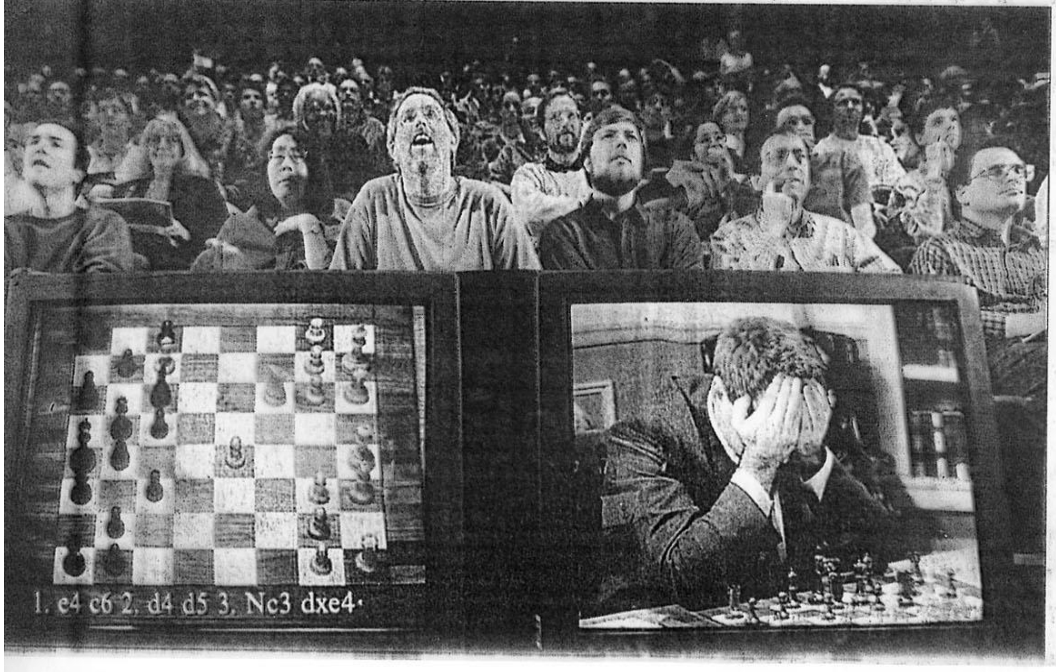
【本報綜合紐約十二日外電報導】超級電腦「深藍」擊敗世界西洋棋王卡斯帕洛夫，不但是一件劃時代的大事，也讓人重新省思人類與電腦之間的關係。不過，卡斯帕洛夫和創造深藍的IBM研究人員對於深藍究竟只是一個龐大的計算機或是一種新的智慧看法互異。

西密西根大學哲學教授麥格魯說，人類一直無法擺脫失去對自己發明物控制權的恐懼，尤其是像深藍這類有思考力的物體。

撰寫有關先進電腦書籍的作家麥柯杜克女士說，人們一直有個迷思，認為下西洋棋和人類智慧的發展息息相關。但是深藍獲勝並不意味它比人類更聰明，只能說這個電腦棋王下了一場精采的棋賽。

加拿大亞伯塔大學電腦學教授薛佛也說，若和醫生以及當醫生所需的技術相比，或是和謝瓦茲卡夫將軍在波斯灣戰爭中的運籌帷幄比較起來，棋術是非常簡單的，只涉及特定數目的物件以及簡單明確的規則。

但卡斯帕洛夫卻無法只把深藍視為計算機。觀察家說，棋王輸在動了情緒，他被對手永不



《深藍的靈魂人物》

「瘋鳥」許峰雄　天才人物

「深藍」一戰成名，經過六場鏖戰，打敗號稱當今最偉大的西洋棋棋王卡斯帕洛夫。「深藍」強大的棋力來自IBM的研究團隊，這個七人小組的靈魂人物則是台大電機系畢業、今年才卅九歲的許峰雄。

「深藍」稱王的原因在於它有超級快速的中央處理器以及特別撰寫的程式，許峰雄就是設計、製造CPU及開發控制程式的主力。

「他實在是一個天才型的人。」台大電機系教授雷欽隆是許峰雄的大學同窗好友，得知電腦打敗棋王時，他的反應與許多計算科學界的人相似，並不感到意外。他說，綽號「瘋鳥」的許峰雄是天才，但也「有一點怪」。

許峰雄在大學時成績就非常好，語文能力也很強，對吸收國外新知有直接幫助，也老是英文歌不離口；但他不喜歡上課。雷欽隆回憶，許峰雄常在上課的時候和同學聊天，聊到別人都快沒有辦法聽課的時候，他會突然問老師一個問題，於是老師「就僵在那裡了」。但他不讀書，也能在電機系一百八十名學生中，搶到第三、四名畢業。

大學時許峰雄就對電腦下棋的研究產生興趣。他的圍棋實力在一段左右

，西洋棋只是
是想教電腦下
是改以西洋棋
式，但當時國
當然也就沒有

大學畢業後
時卡內基美倫
究院院士孔祥
念。卡內基美
坦福並稱美國
峰雄是該校大
個給獎學金的

到了美國，
峰雄的興趣也
所的時候，他
的電腦，這是
思」的計算能
步，只是「深
經打敗過不少
九四年贏得世
震驚電腦界與
界談到電腦下
就是「NO.1」

今年初，許
系演講的時候
今年二度與卡
贏。如今預言
樂得再度歡唱

AI Bible 的封面

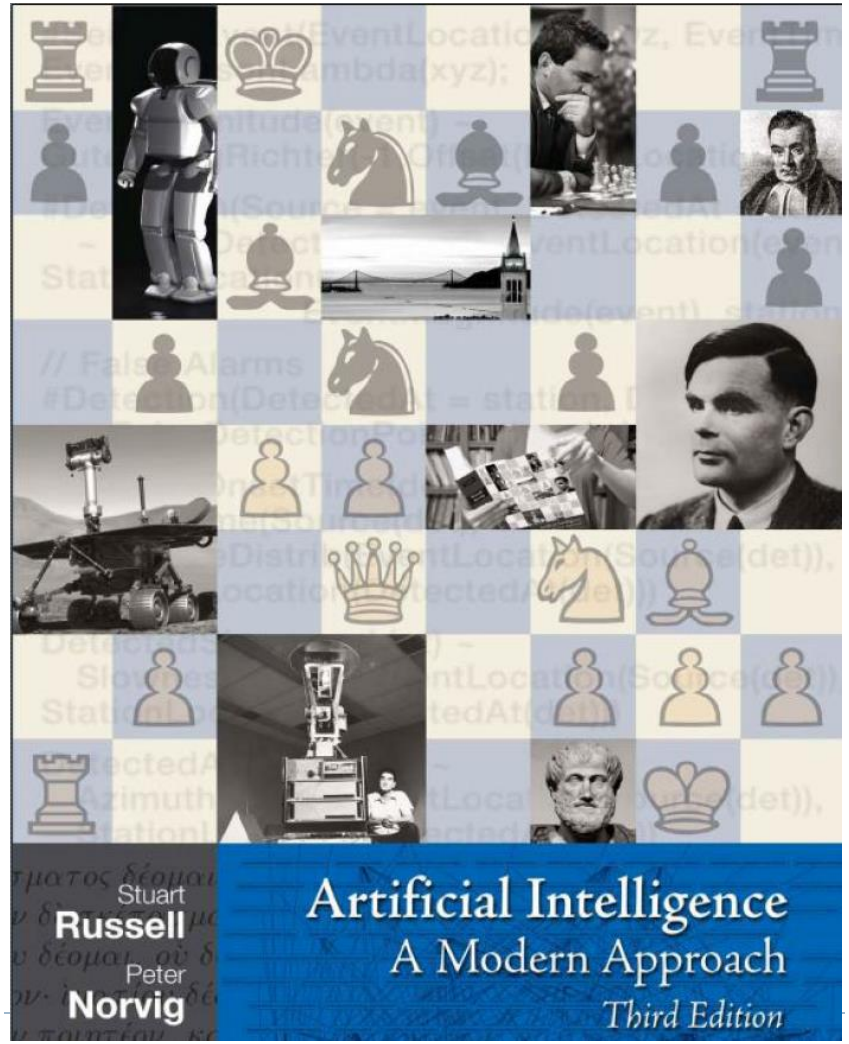
- 1997年比賽
  - 深藍致勝一局(第六局)的盤面

書名(Title):

Artificial Intelligence: A Modern Approach

作者(Authors):

S. Russell and P. Norvig

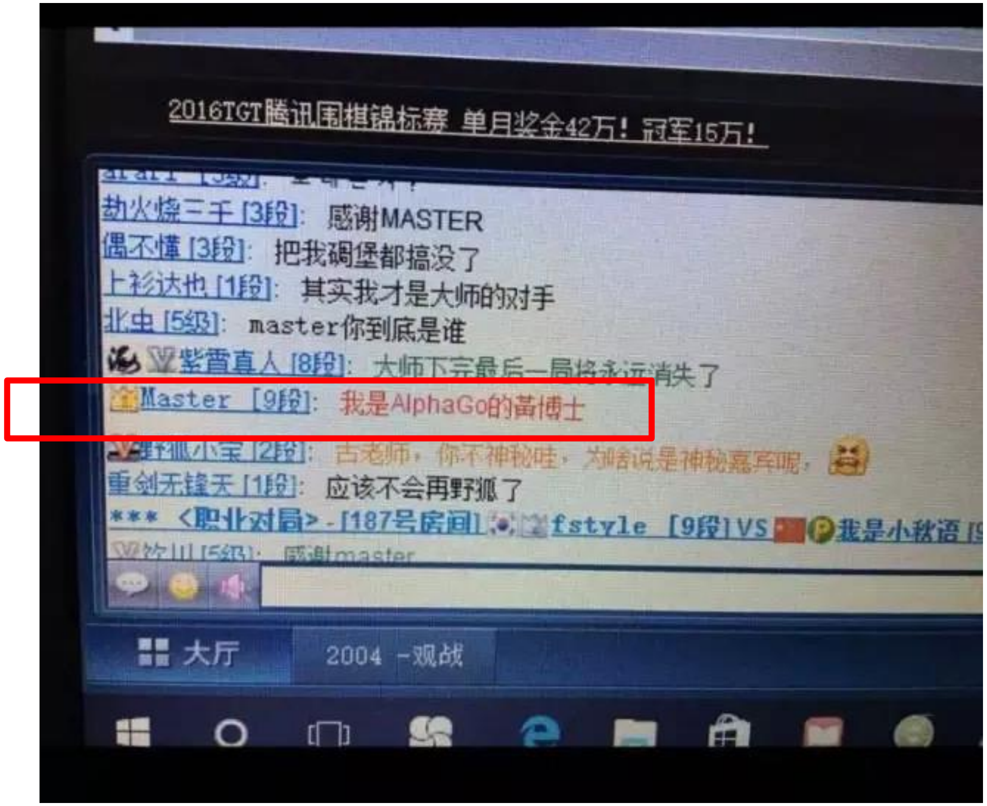


AlphaGo vs. 李世石

- 4:1 (left: AlphaGo (Aja), right:李世石)

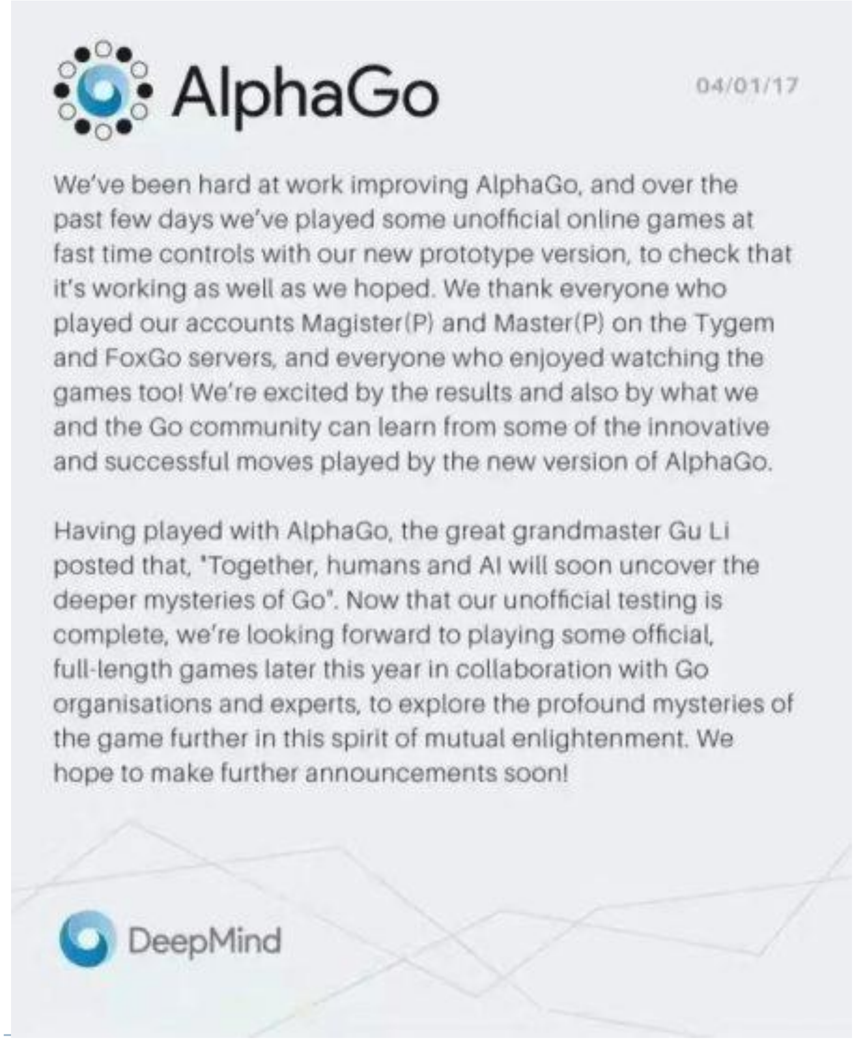
Master Beat Go Champions/Grand Masters on 2016/12/30 ~ 2017/1/4 (not official)

60 (master) : 0 (human)



2016TGT腾讯围棋锦标赛 单月奖金42万！冠军15万！

劫火烧三千 [3段]： 感谢MASTER
偶不懂 [3段]： 把我碉堡都搞没了
上衫达也 [1段]： 其实我才是大师的对手
北虫 [5级]： master你到底是谁
紫霄真人 [8段]： 大师下完最后一局将永远消失了
Master [9段]： 我是AlphaGo的黄博士
野狐小宝 [2段]： 古老师，你不神秘哇，为啥说是神秘嘉宾呢，
重剑无锋天 [1段]： 应该不会再野狐了
*** <职业对局> - [187号房间] fstyle [9段] VS 我是小秋语 [9
饮山 [5级]： 感谢master



AlphaGo 04/01/17

We've been hard at work improving AlphaGo, and over the past few days we've played some unofficial online games at fast time controls with our new prototype version, to check that it's working as well as we hoped. We thank everyone who played our accounts Magister(P) and Master(P) on the Tygem and FoxGo servers, and everyone who enjoyed watching the games too! We're excited by the results and also by what we and the Go community can learn from some of the innovative and successful moves played by the new version of AlphaGo.

Having played with AlphaGo, the great grandmaster Gu Li posted that, "Together, humans and AI will soon uncover the deeper mysteries of Go". Now that our unofficial testing is complete, we're looking forward to playing some official, full-length games later this year in collaboration with Go organisations and experts, to explore the profound mysteries of the game further in this spirit of mutual enlightenment. We hope to make further announcements soon!

DeepMind

人機圍棋最終決戰 – The Future of Go Summit

- AlphaGo vs. 柯潔(世界排名第一): 3:0

人機圍棋最終決戰－團體賽

- AlphaGo (win) vs. 陳耀燁、周睿羊、芈昱廷、時越、唐韋星（均為9段，曾獲世界冠軍）

AlphaGo的衝擊

不只圍棋界、電腦圍棋

對整個**人工智慧領域**,

甚至整個**計算機領域, 人類社會**

影響層面極大!!

遠勝深藍的影響!!!

Impact of AlphaGo vs. Deep Blue

- When compared with Deep Blue,
  - "Not much Go domain knowledge is used."
  - A big gap to beat human Go champions which most people thought a decade away.

➔

  - More inspiration
  - Higher impact

AlphaGo的成功

- 採用許多general machine learning techniques
  - Deep learning (DL, 深度學習)
  - Reinforcement learning (RL, 強化式學習)
  - Combine DL+RL
    - Called Deep Reinforcement Learning (DRL)

  註: 無需太多圍棋知識.
- 許多頂尖深度學習科學家
- Google的大量計算支援

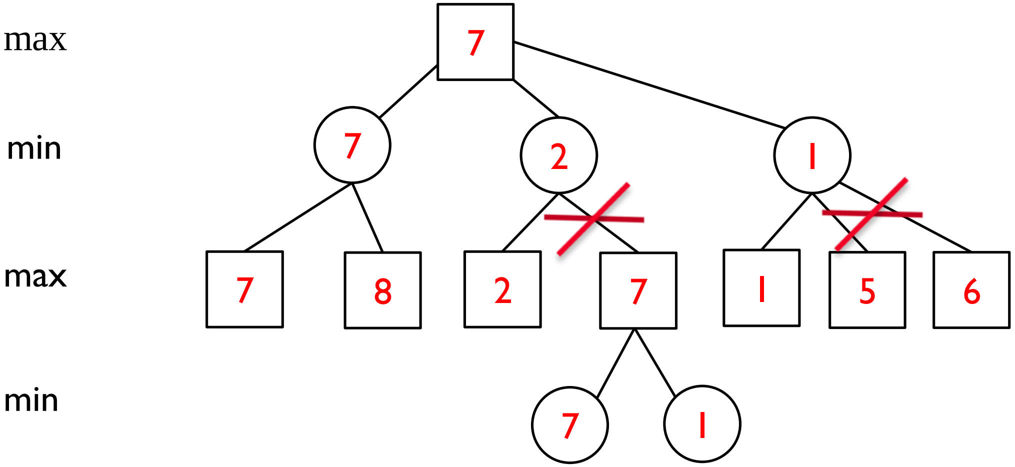
Approaches to Computer Games

Approaches to Computer Games (I)

- Tree search
  - Alpha-beta search
    - 六子棋(Connect6), 象棋
  - Backtracking
    - Nonogram, NuriKabe
  - Expectiminimax search
    - 2048, 暗棋, 麻將
  - Proof number search
    - 六子棋, 圍棋
  - Threat space search
    - 六子棋
  - Combinatorial game theory
    - NoGo
  - Monte-Carlo tree search
    - 圍棋, NoGo,愛因斯坦棋

Alpha-Beta Search (Traditional)

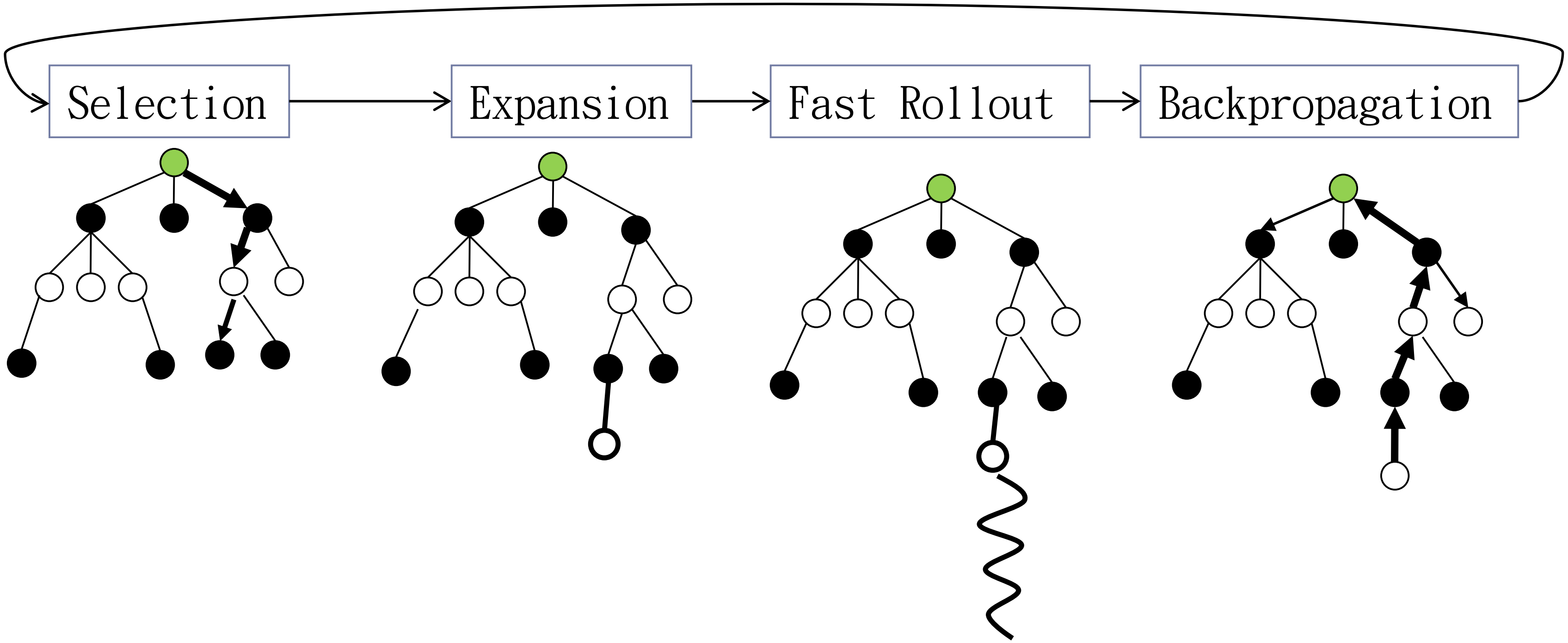
- Alpha-beta pruning
  - Greatly reduce tree sizes from $O(b^d)$ to $O(b^{d/2})$
- Comments:
  - A method dominating computer games for 4-5 decades.
  - Perform very well for chess and Chinese chess, not for Go.
  - Need experts to evaluate leaves (policy vs. value)



Monte-Carlo Tree Search (MCTS)

(Modern)

- Also a kind of (model-free) **Reinforcement learning**
- Perform well for Go, and many other games,
  - Other games like Havannah, Hex, GGP (General Game Playing.
  - Even many other applications, like mathematical optimization problems,
    - Scheduling, UCP, camera coverage.



Approaches to Computer Games (II)

- Machine Learning
  - Deep Learning：圍棋、NoGo
    - Supervised Learning Policy Network
    - Value network
  - Reinforcement Learning (RL)
    - Monte-Carlo learning (including MCTS)：圍棋, NoGo
    - Temporal Difference (TD) learning ：2048, 六子棋,愛因斯坦棋, Othello
  - Deep Reinforcement Learning (DRL) ：圍棋
    - Reinforcement Learning Policy Network
  - Other Machine Learning
    - Comparison Training：象棋
- Learning Networks
  - N-tuple：2048, 愛因斯坦棋, Othello
  - Deep Convolutional Neural Network (DCNN) ：圍棋、NoGo、麻將

深度學習
Deep Learning

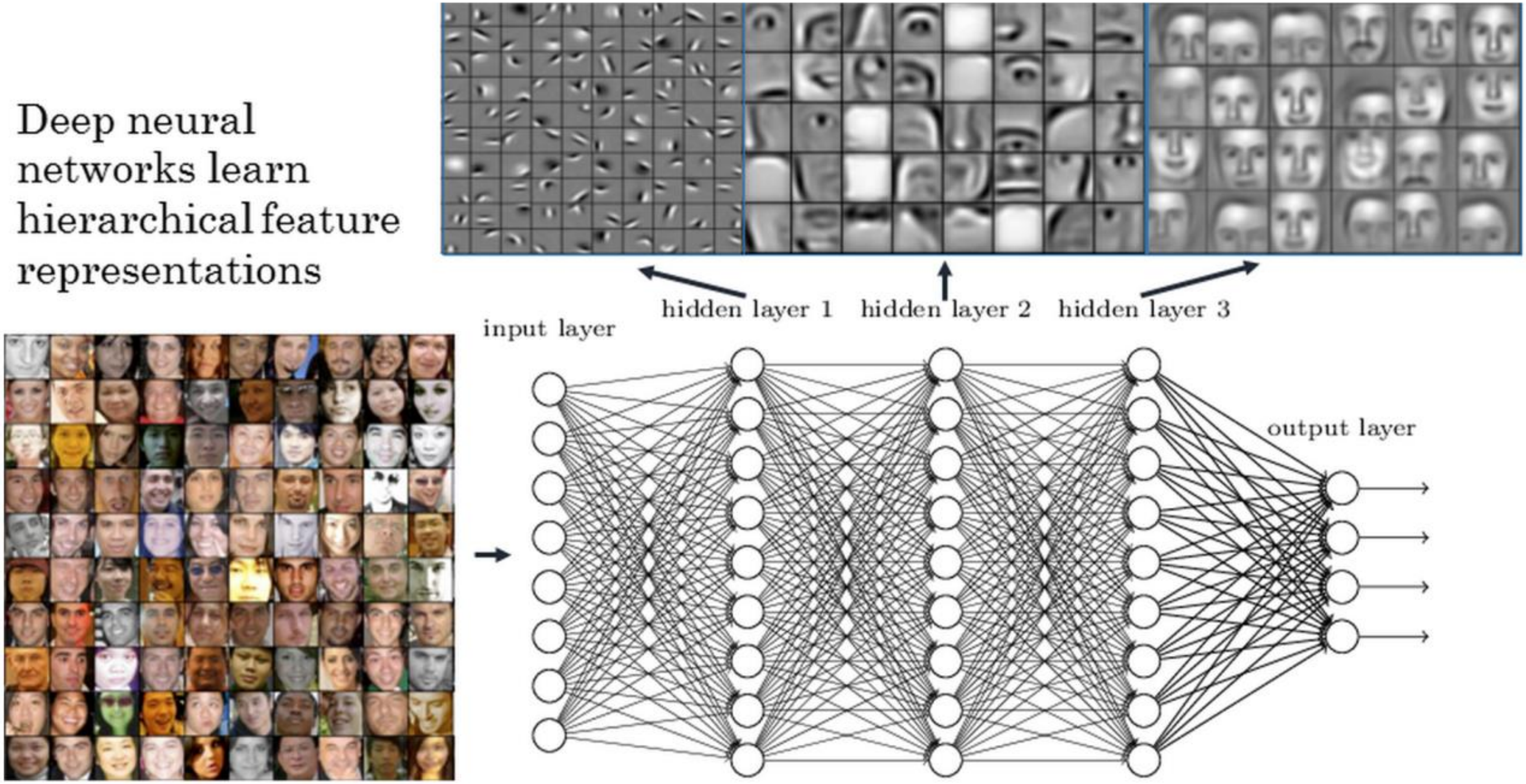
深度強化式學習 ▶

Deep Learning

- Deep neural network (DNN)
  - Deep convolutional neural networks (DCNN)
  - Recurrent neural networks (RNN)
  - Long short term memory (LSTM)
  - Generative Adversarial Networks (GAN)
  - Many other networks:
    - LeNet-5, AlexNet, ZFNet, Network in Network, VGG Network, GoogLeNet, Dual Path Network, Squeeze-and-Excitation Networks, Residual Networks, ResNeXt, DenseNet.

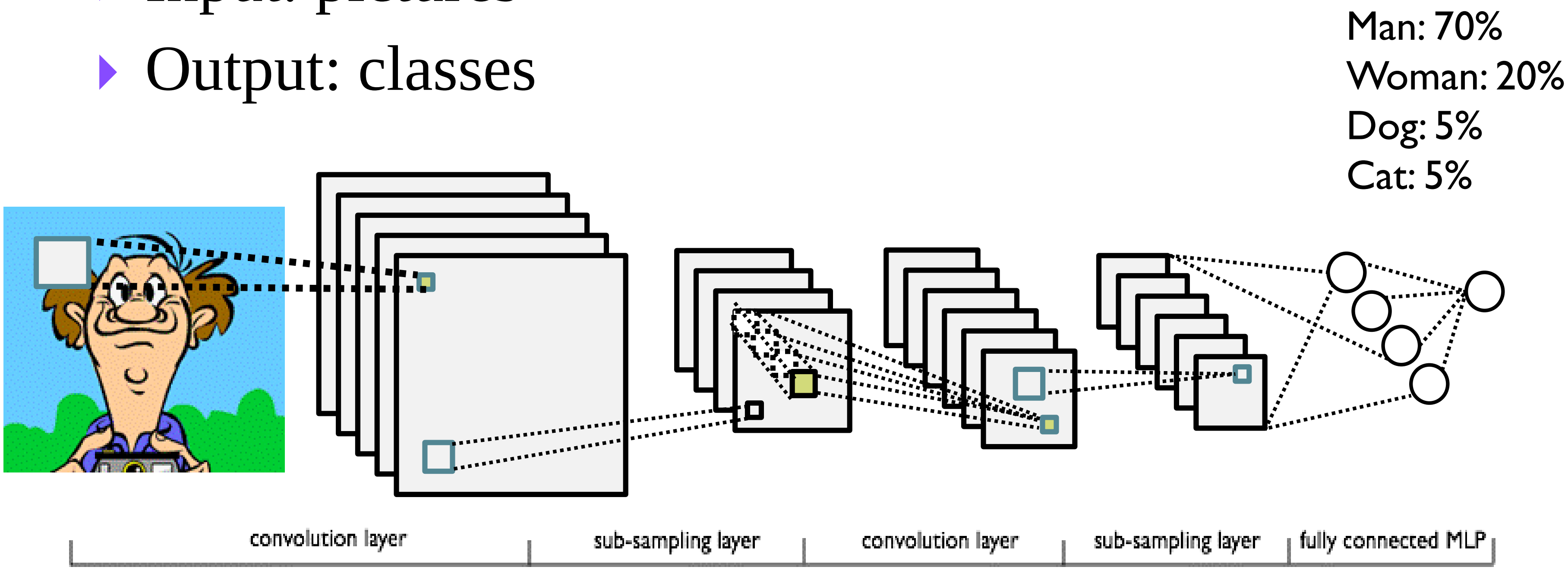
Deep Neural Network (深度類神經網路)

Deep neural networks learn hierarchical feature representations



Deep Convolutional Neural Networks (DCNN; 深度卷積類神經網路)

- 辨識圖片 Recognize patterns (分類器Classifier)
  - Input: pictures
  - Output: classes

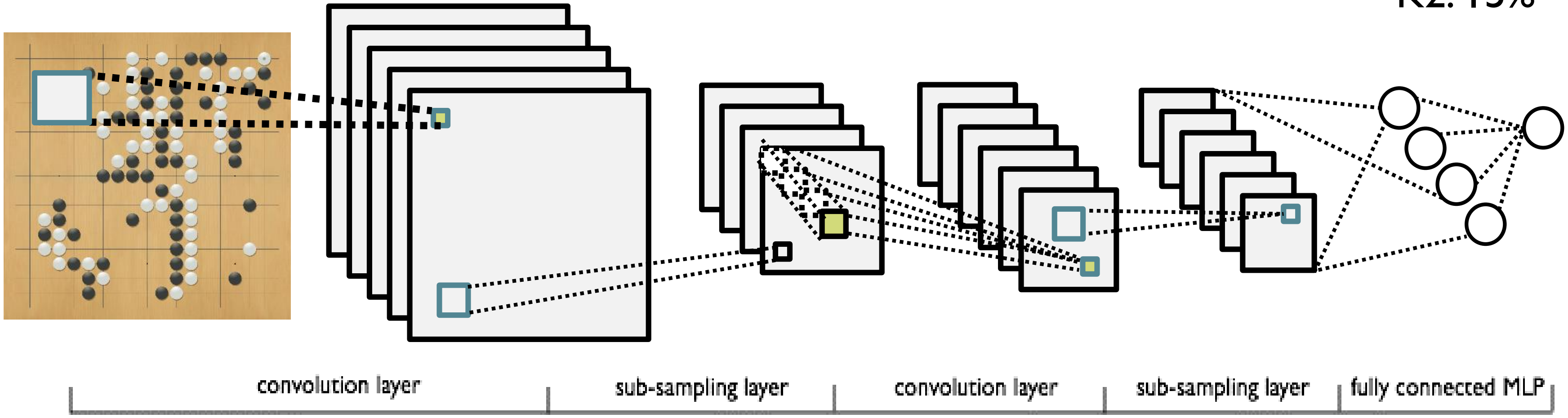


Man: 70%
Woman: 20%
Dog: 5%
Cat: 5%

Deep Convolutional Neural Networks (DCNN; 深度卷積類神經網路)

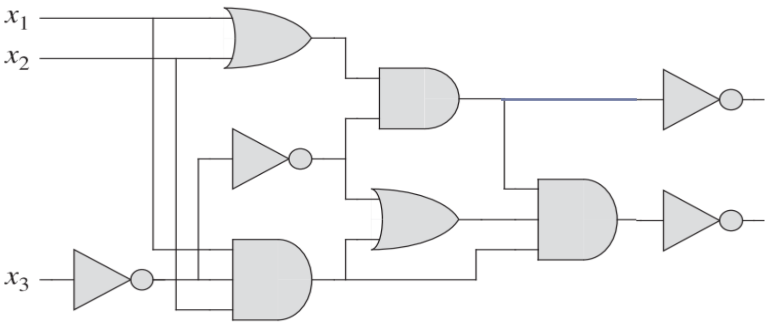
- 辨識棋型預測高手著手(Classifier)
  - Input: boards
  - Output: which moves (or values of boards).

G11: 40%
H2: 25%
D5: 20%
K2: 15%

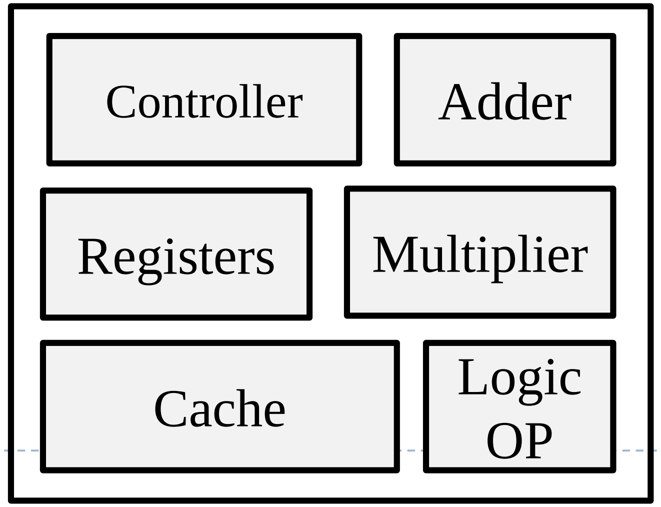


Why Deep? (1/2)

- Recall Circuit Design:
  - 理論上只要兩層gates即可表示所有涵數
    - E.g., all $y_i = (\sim x_1+x_2+\sim x_3+\ldots) * (x_2+\sim x_4+x_7+\ldots) * \ldots$
  - 但實際上非常少電路設計, 只用兩層!!
  - Why?
    → 反而更複雜
    Exponentially grow!!
    - SAT is NP-complete



- Solution:
  - Use more layers (or blocks) to squeeze the size.

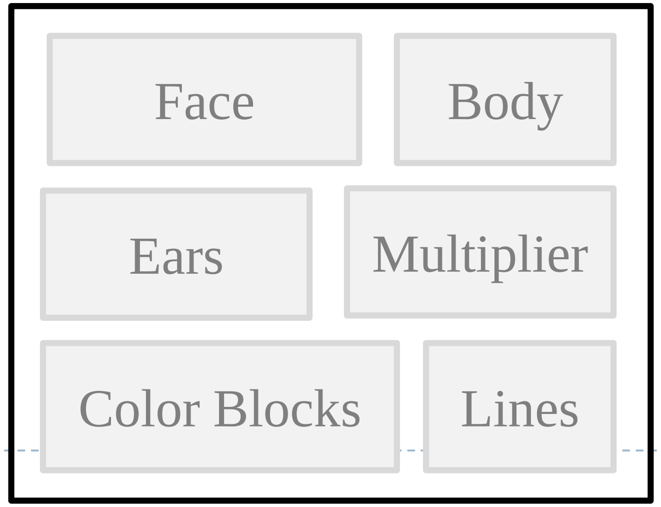


Why Deep? (2/2)

- 對NN, 理論上只要用三層(3 layers)即可表達所有cases
  - Input Layer
  - Hidden Layer
  - Output Layer
- Problem: (Similar to Circuit Design)
  - Hidden Layer need to grow exponentially.
- Solution:
  - Use deeper layers.

- But, a new problem is:
  - Not to be too deep and too wide.
    - Prevent from overfitting.
  - Squeeze into a DNN carefully.

This is KNOW-HOW!!!



Face

Body

Ears

Multiplier

Color Blocks

Lines

成功的故事

- 2012年國際著名賽事ImageNet比賽中， Hinton實驗室團隊採用了深度學習獲勝，失敗率僅15%。
  - **過去的獲勝者失敗率約為25%。(25% → 15%)**
  - 2013年ImageNet比賽所有參賽者都採用深度學習.
- 2012年, Google實驗室研究者從YouTube視頻中選取了大約一千萬張靜態圖片，並由Google Brain訓練三天后，能辨識出人臉、身體等，還有貓！
- 2012年, Hinton 的學生 Dahl 就應用DL技術，打敗了默克（Merck）藥廠現行的系統，成功提高了對特定化學分子間反應的預測力，以便更有效率地找出有用的藥物。他的團隊藉由這個深度學習系統，提升了約15%的預測力，更獲得了默克藥廠懸賞的2.2萬美金獎金。
- 華盛頓大學的 William Stafford Noble 也應用深度學習系統來預測胺基酸鏈會組成如何的蛋白質、並可進一步預測此蛋白質的性狀。2015年已有著將近10萬個蛋白質結構的全球資料庫。
- 麻省理工學院的 Sebastian Seung 也利用深度學習來分析腦部切片、以建立三維空間的腦圖，以及神經束的走向。
- Rob High（位於德克薩斯州的公司首席技術官）聲稱，他們已經嘗試性地使用深度學習，提升IBM Watson的模式識別能力。
- 新創公司(Behold.ai)運用DL，迅速診斷出乳癌等疾病.
- 2016-2017, AlphaGo擊敗李世石高段棋士.

(深度)強化式學習
(Deep) Reinforcement Learning

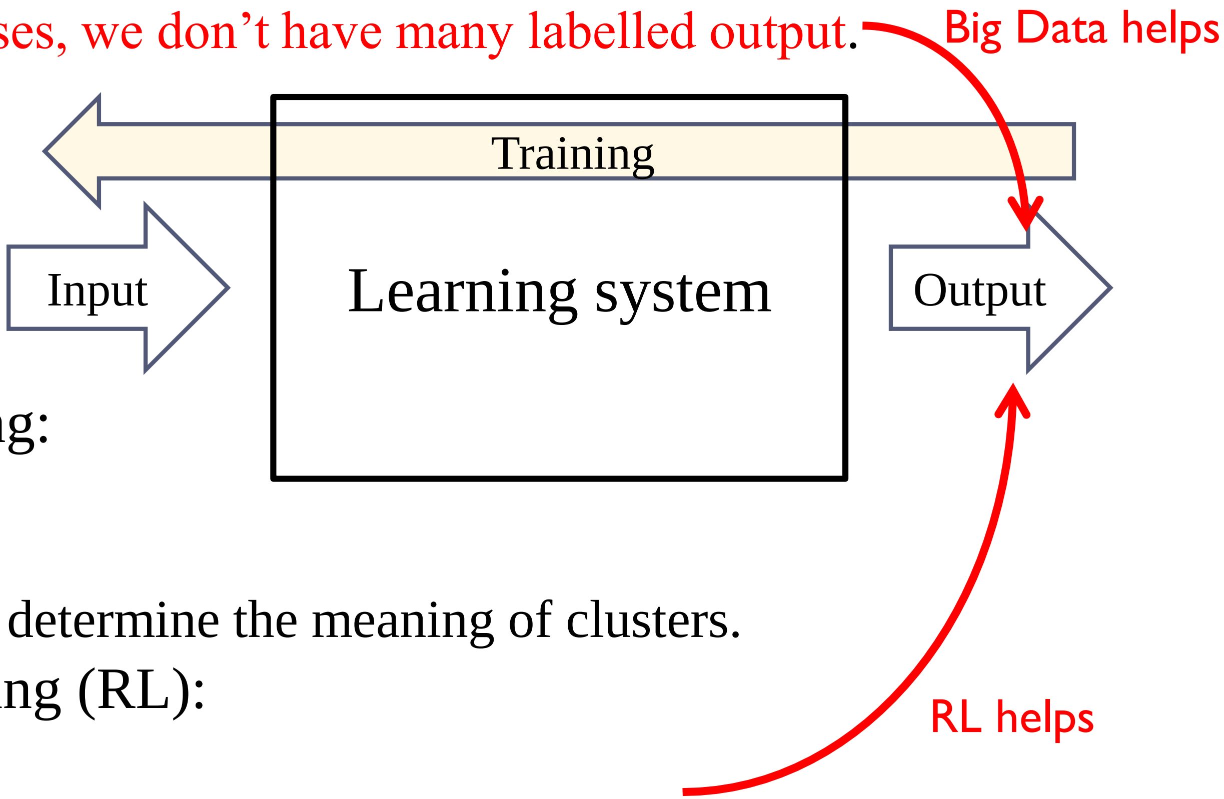
案例研究 ▶

Successful Examples

- In AI, it has been used to defeat human champions at games of skill.
  - Backgammon (Tesauro, 1994).
  - Connect6/2048/Threes! (Wu et al., 2015). Reach the top levels.
  - Go, used in the past 10 years. (Monte-Carlo Tree Search)
  - AlphaGo and Atari games (Deep Reinforcement Learning)
- In robotics, to fly stunt maneuvers in robot-controlled helicopters (Abbeel et al., 2007).
- In neuroscience it is used to model the human brain (Schultz et al., 1997);
- In psychology to predict animal behavior (Sutton and Barto, 1990).
- In economics, it is used to understand the decisions of human investors (Choi et al., 2007), and to build automated trading systems (Nevmyvaka et al., 2006)
- In engineering, it has been used to allocate bandwidth to mobile phones and to manage complex power systems (Ernst et al., 2005).

Differences from Other Learnings

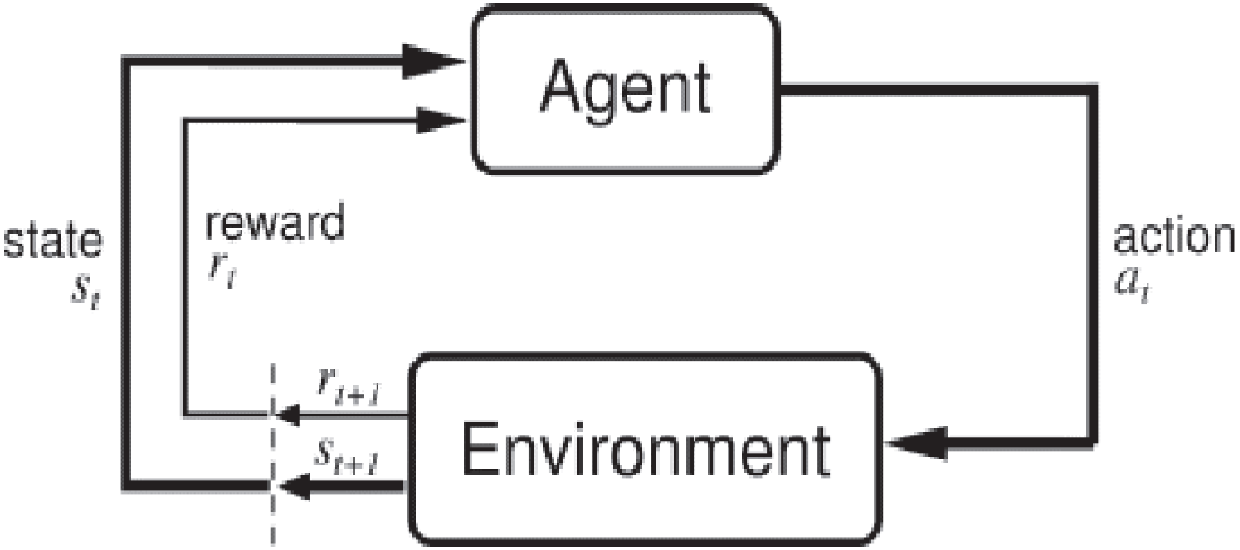
- Supervised Learning:
  - For each given input, give labelled output.
  - Problem: In many cases, we don't have many labelled output.



- Unsupervised Learning:
  - No labelled output.
  - Cluster on your own.
  - Problem: it is hard to determine the meaning of clusters.
- Reinforcement Learning (RL):
  - No labelled output.
  - But, for MDP, we can make use of reward to learn. ← this is the key.

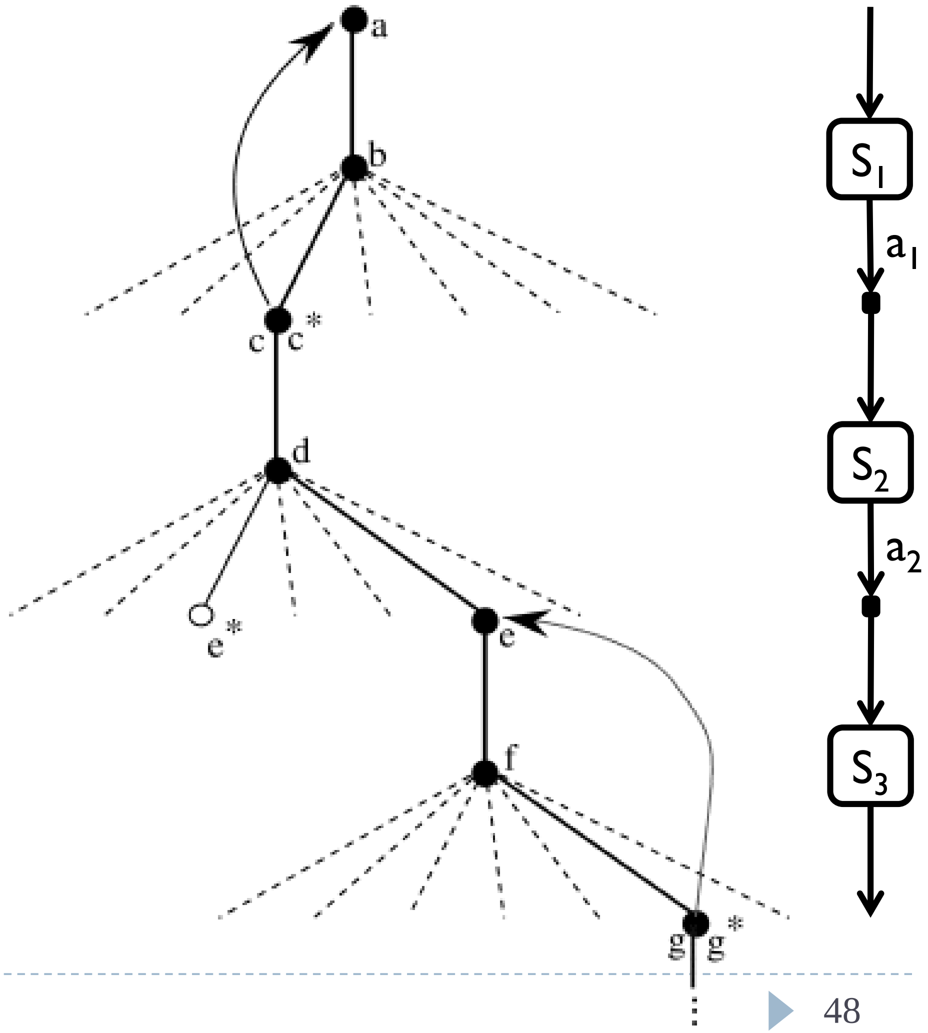
Reinforcement Learning (RL)

- A computational approach to learning from interaction
  - MDP (Markov Decision Processes)
- Model-free
- Agent-Environment Interaction Framework



States and Actions in the Framework

Environment:	reaction
Agent:	action
Environment:	reaction
Agent:	action
Environment:	reaction
Agent:	action



Go

Environment: opponent's move

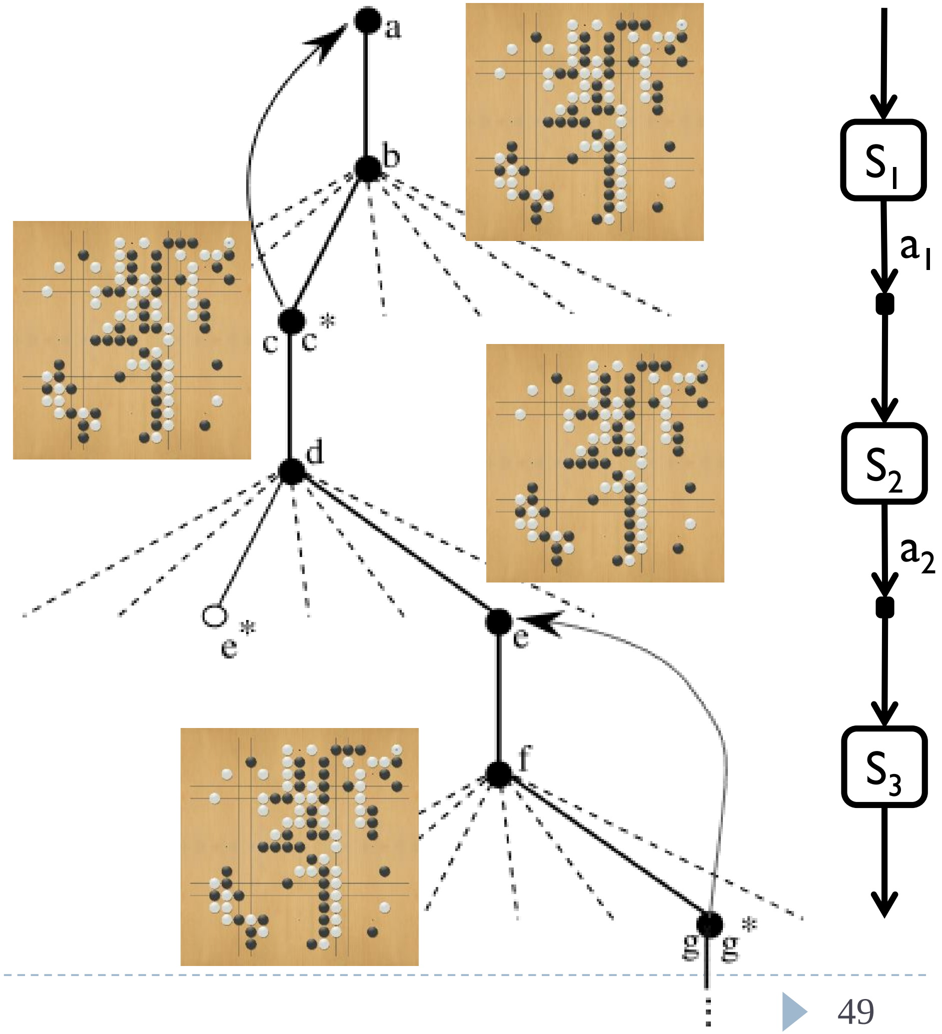
Agent: our move

Environment: opponent's move

Agent: our move

Environment: opponent's move

Agent: our move



2048

Environment: Tile generation {

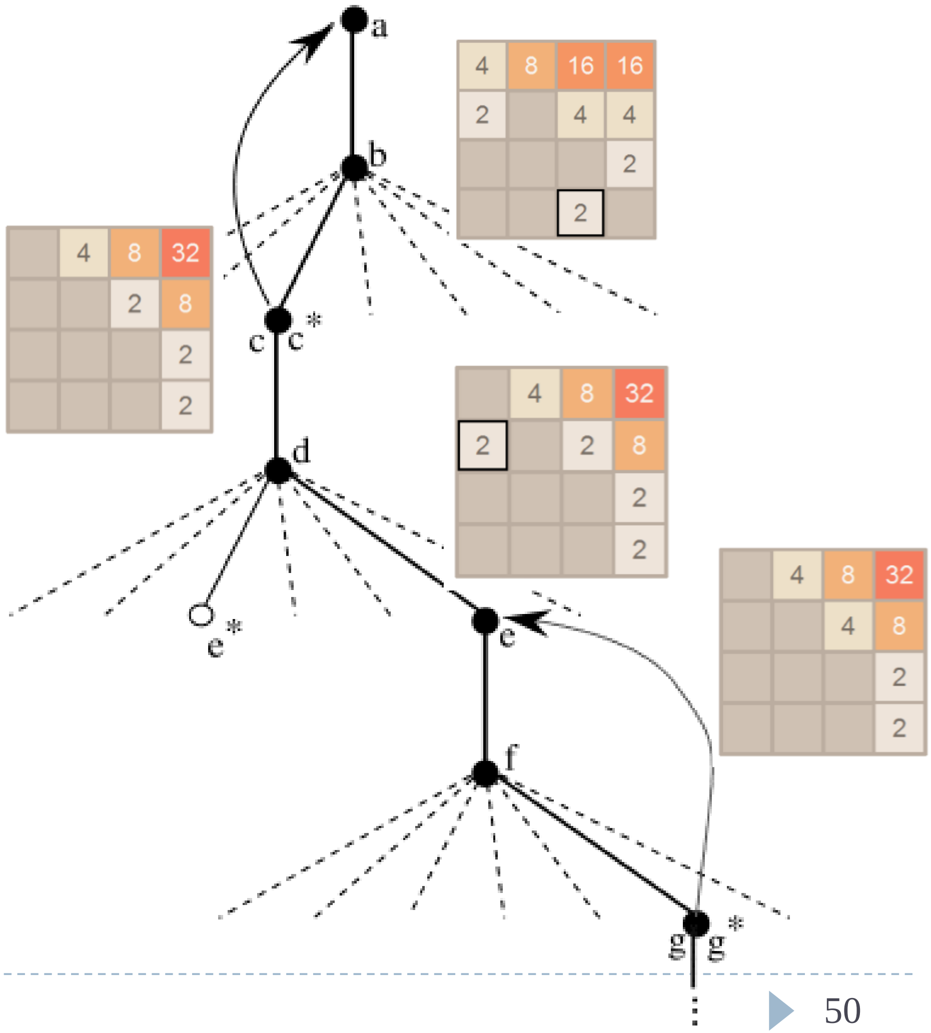
Agent: our move {

Environment: Tile generation {

Agent: our move {

Environment: Tile generation {

Agent: our move {



4	8	16	16
2		4	4
			2
		2	

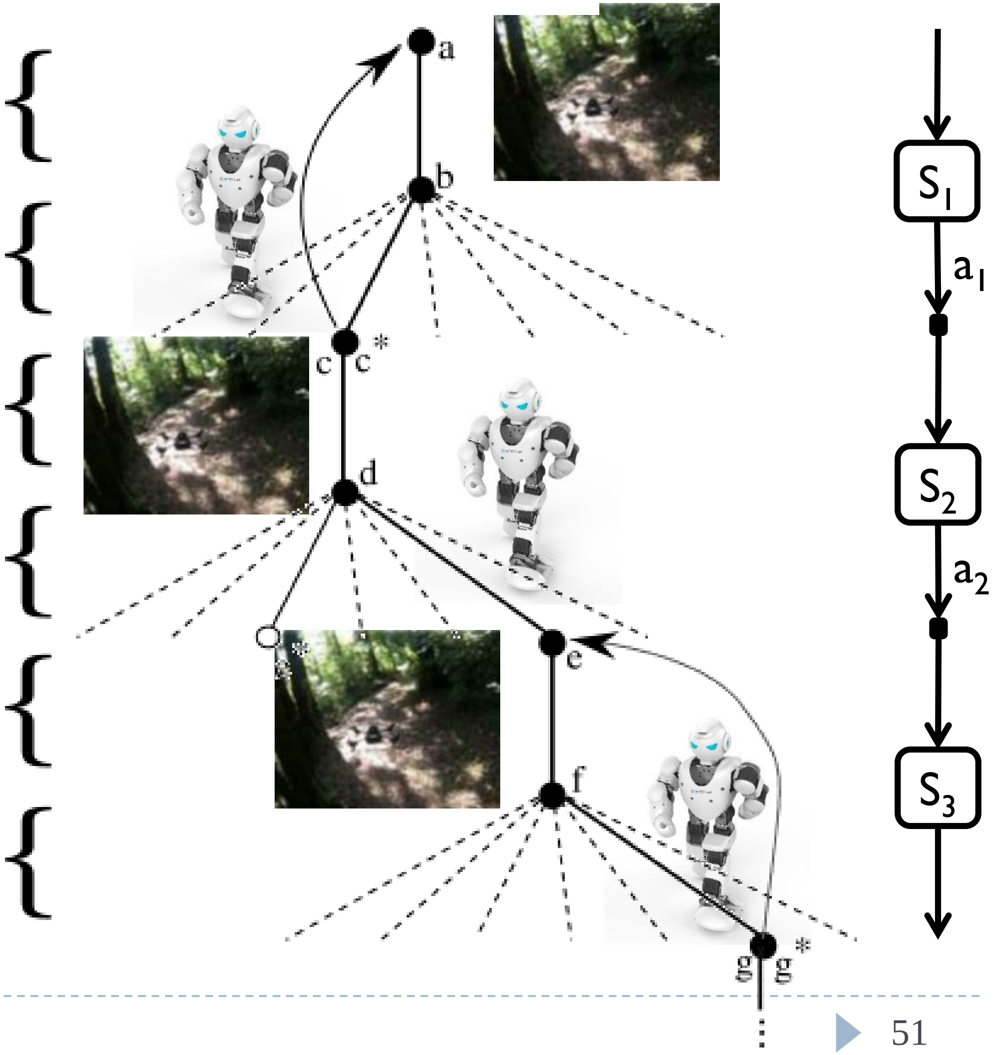
	4	8	32
		2	8
			2
			2

	4	8	32
2		2	8
			2
			2

	4	8	32
		4	8
			2
			2

Robot

Environment:	Dynamics
Agent:	Navigate
Environment:	Dynamics
Agent:	Navigate
Environment:	Dynamics
Agent:	Navigate



Two Model-Free Reinforcement Learning

- Monte-Carlo Learning
- Temporal-Difference (TD) Learning

What is model-free?

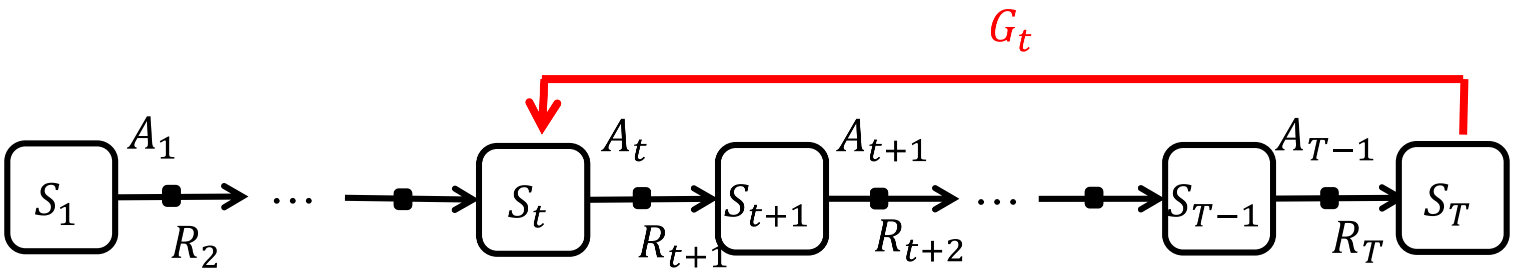
- Do not depend on environment!!!
  - For example, no need to know the response rules of 2048!

Monte-Carlo Learning

- Incremental Monte-Carlo
  - Update value $V(S_t)$ toward actual return G_t

$$V(S_t) \leftarrow V(S_t) + \alpha(G_t - V(S_t))$$

  - α: learning rate, or called step size.
- Unbiased, but high variance.

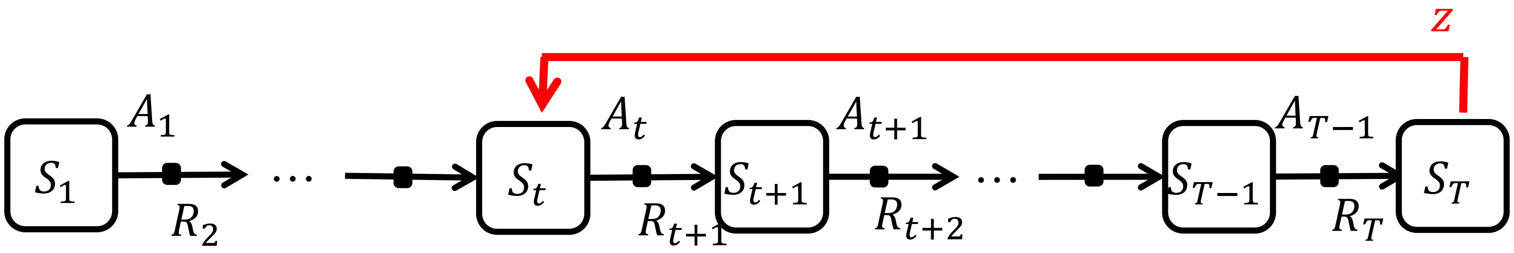


Example: AlphaGo

- Use stochastic policy gradient ascent to maximize the likelihood of the human move a selected in state s

$$\Delta\theta = \alpha \nabla_\theta \log \pi_\theta(s_t, a_t) \cdot z$$

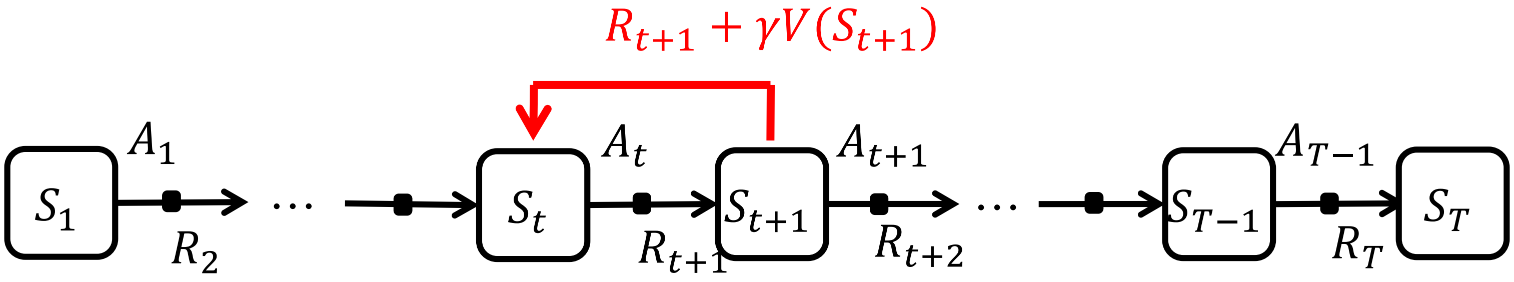
  - θ: network parameter.
  - α: learning rate
  - z: the value of the episode
    - win/loss (1/-1) of the game



Temporal-Difference (TD) Learning

- Simplest temporal-difference learning algorithm: TD(0)
  - Update value $V(S_t)$ toward estimated return $R_{t+1} + \gamma V(S_{t+1})$

    $$V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$$
  - TD target: $R_{t+1} + \gamma V(S_{t+1})$
  - TD error: $R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$
- Biased, but lower variance



案例研究
Case Studies

2048

Go (圍棋)

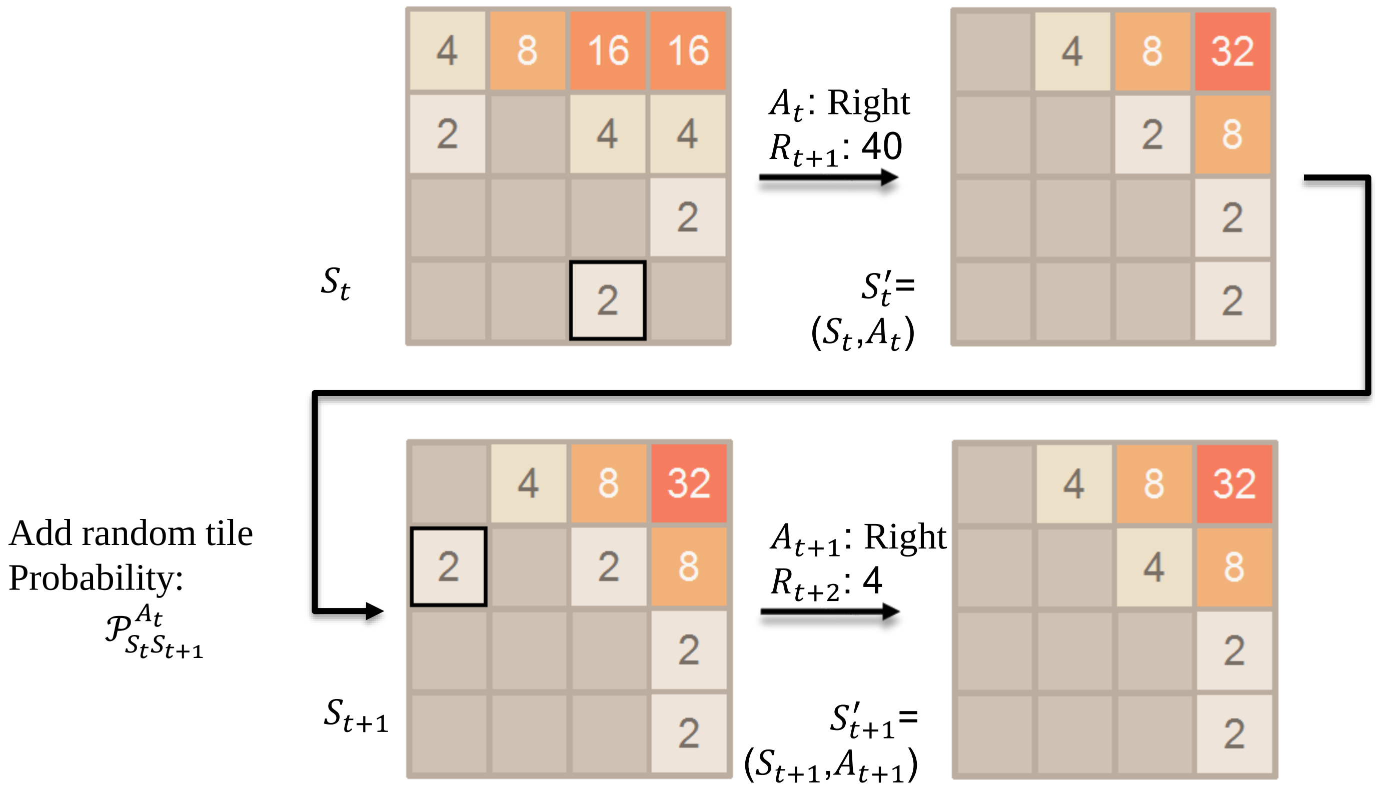
Video Games

Robotics

案例研究：圍棋 ▶

TD Learning: 2048

- [Szubert et al., 2014; Yeh et al., 2016]



S_t

4	8	16	16
2		4	4
			2
		2	

A_t: Right
R_{t+1}: 40

S'_t= (S_t,A_t)

	4	8	32
		2	8
			2
			2

Add random tile
Probability:
$\mathcal{P}_{S_t S_{t+1}}^{A_t}$

S_{t+1}

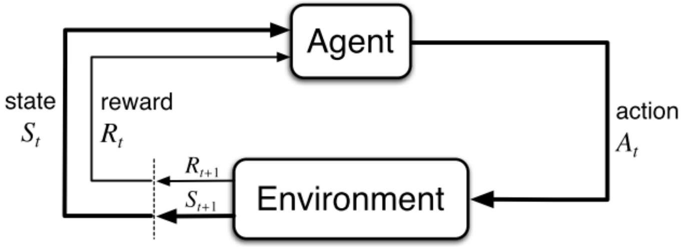
	4	8	32
2		2	8
			2
			2

A_{t+1}: Right
R_{t+2}: 4

S'_{t+1}= (S_{t+1},A_{t+1})

	4	8	32
		4	8
			2
			2

2048 RL Agent

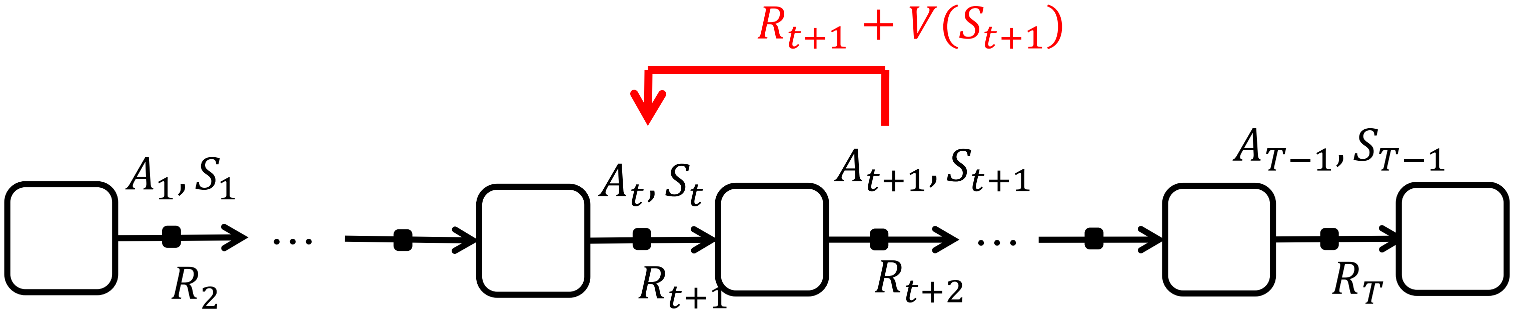


- Value function:
  - The expected score/return G_t from a board S
  - But, #states is huge
    - About 17^{16} ($=10^{20}$).
      - Empty, 2 ($=2^1$), 4 ($=2^2$), 8 ($=2^3$), …, 65536 ($=2^{16}$).
  - Need to use value function approximator.
- Policy:
  - Simply choose the action (move) with the maximal value based on the approximator.
- Model: agent's representation of the environment
  - After a move, randomly generate a tile:
    - 2-tile: with probability of 9/10
    - 4-tile: with probability of 1/10
  - Reward: simply follow the rule of 2048.

TD Learning in 2048

- State (afterstate) value function: (Normally $\gamma = 1$)
  - Update value $V(S_t)$ toward TD target $R_{t+1} + \gamma V(S_{t+1})$

$$V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$$



A kind of Q-Learning

Linear Value Function Approximation

- Represent value function by a linear combination of features of n-tuples

$$\hat{v}(S;\theta) = x(S)^{\mathrm{T}}\theta = \sum_{j=1}^{n} x_j(S)\theta_j$$

- Gradient of $\hat{v}(S,\theta)$:

$$\nabla_\theta \hat{v}(S,\theta) = x(S)$$

Gradient Descent

- Update value $V(S_t)$ towards TD target $y_t = R_{t+1} + V(S_{t+1})$

$$\Delta V = (R_{t+1} + V(S_{t+1}) - V(S_t)) = (y_t - V(S_t))$$

$$V(S_t) \leftarrow V(S_t) + \alpha \Delta V$$

  - α: learning rate, or called step size.
  - Note: $\gamma = 1$ here.
- Objective function is to minimize the following loss in parameter θ. (note: $\hat{v}(S, \theta) = x(S)^{\mathrm{T}}\theta$)

$$\mathcal{L}(\theta) = \mathbb{E}\left[\left(y_t - \hat{v}(S, \theta)\right)^2\right]$$

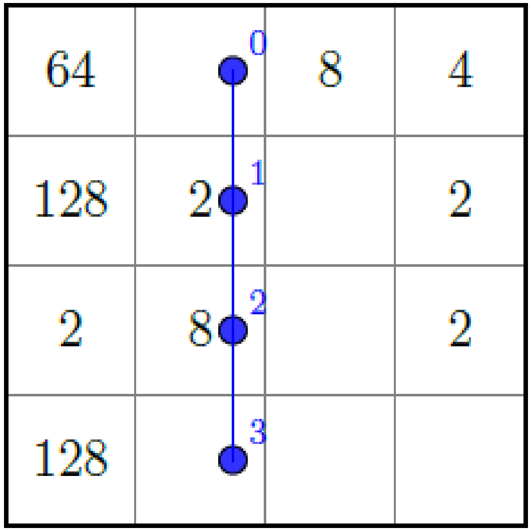
$$\nabla_\theta \mathcal{L}(\theta) = \left(y_t - \hat{v}(S, \theta)\right) \cdot \nabla_\theta \hat{v}(S, \theta) = \Delta V \cdot x(S)$$

- Update features w: step-size * prediction error * feature value

$$\theta \leftarrow \theta + \alpha \Delta V \cdot x(S)$$

N-Tuple Network

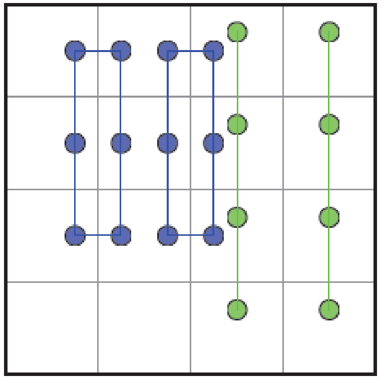
- Example: 4-tuple networks as shown.
  - Each cell has 16 different tiles
  - 16^4 features for this network.
    - But only one is on, others are 0.
    - So, we can use table lookup to find the feature weight.



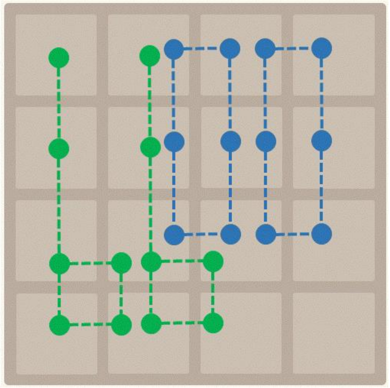
0123	weight
0000	3.04
0001	−3.90
0002	−2.14
⋮	⋮
0010	5.89
⋮	⋮
0130	-2.01
⋮	⋮

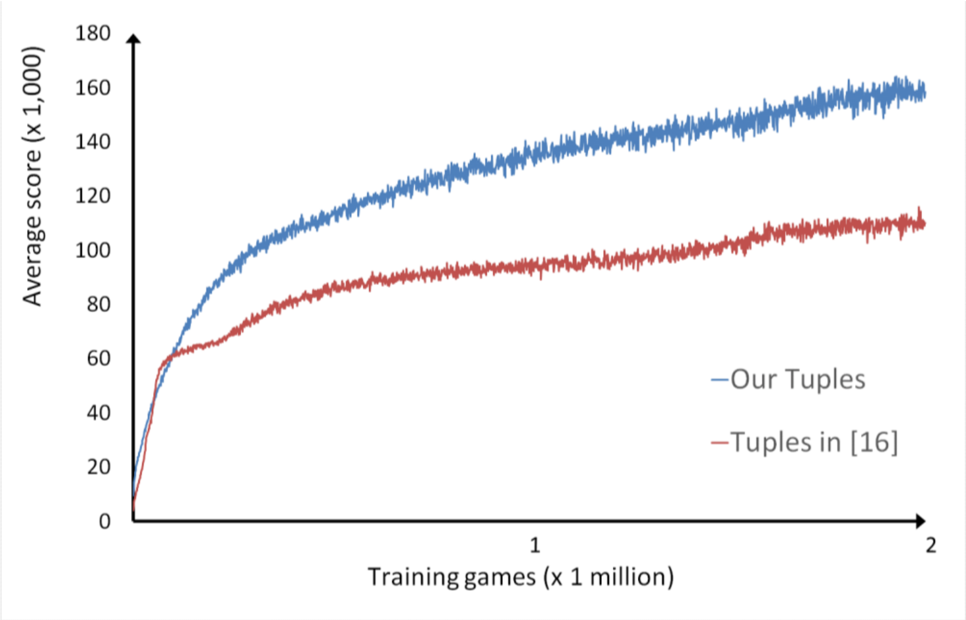
The N-Tuple Networks Used

- [Szubert and Jaskowaski 2014]



- Ours: [Yeh et al., 2016]





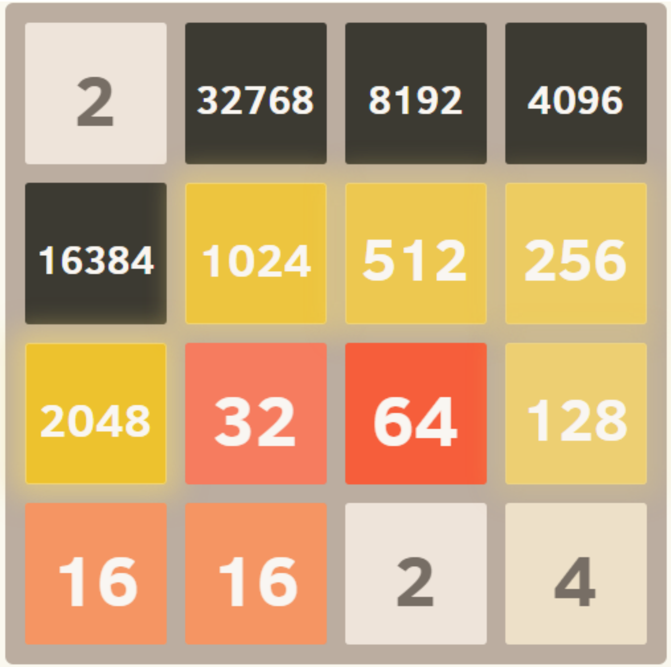
Our Method for 2048 AI

- Also Propose a new TD learning method,
    - Called **Multi-Stage TD Learning**:
    - Split the learning into different game stages!
        - Example of 3-stage Multi-Stage TD learning:
            1. Before 16384-tile.
            2. Before 16384-tile+8192-tile.
            3. After 16384-tile+8192-tile.
- Use 6-Tuple networks
- Incorporate the expectimax search.
- Other tunings:
    - TD-lambda.
    - More features.
- Applied to other games: 愛因斯坦棋, Connect6

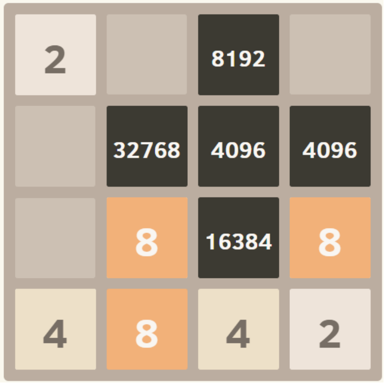
Our Results (2015)

	CGI-2048 (2nd in contest) (100 games)	Kcwu (1st in contest) (100 games)	Xificurk's Program (246 games)	Current CGI-2048 (1000 games)
2048	100.0%	100.0%	**100.0%**	**100.0%**
4096	100.0%	100.0%	**100.0%**	**100.0%**
8192	94%	96%	**99.1%**	**99.5%**
16384	59%	67%	**92.7%**	**93.6%**
32768	0%	2%	**31.7%**	**33.5%**
Max score	367956	625260	**829300**	**833300**
Avg score	251794	277965	**442419**	**446116**
Speed	500 moves/sec	>100 moves/sec	**2-3 moves/sec**	**500 moves/sec**

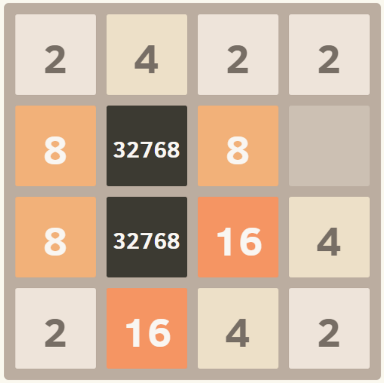
The First 65536



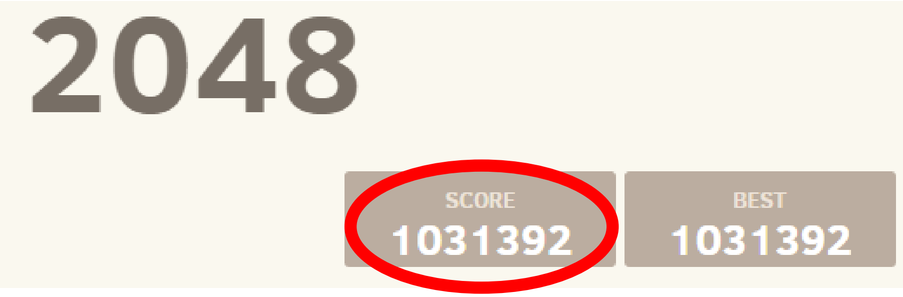
2	32768	8192	4096
16384	1024	512	256
2048	32	64	128
16	16	2	4



2		8192	
	32768	4096	4096
	8	16384	8
4	8	4	2



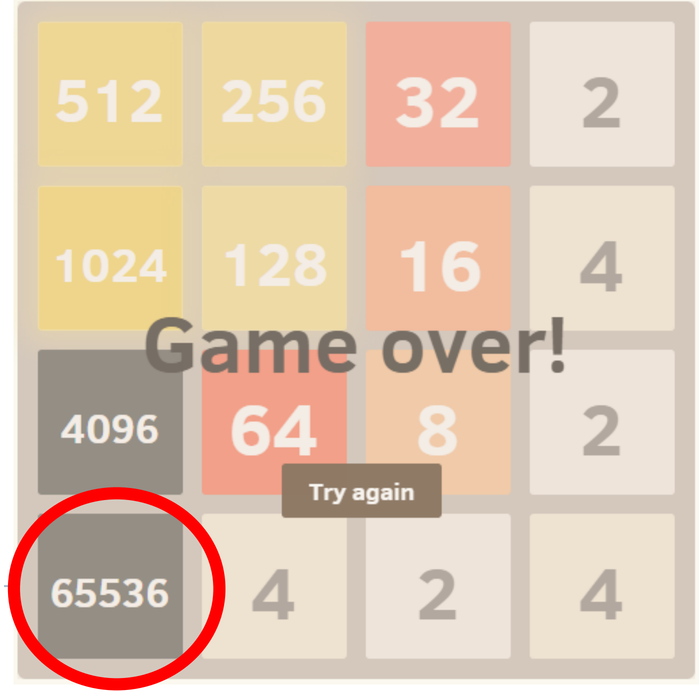
2	4	2	2
8	32768	8	
8	32768	16	4
2	16	4	2



2048

SCORE
1031392

BEST
1031392



512	256	32	2
1024	128	16	4
4096	64	8	2
65536	4	2	4

Game over!

案例研究
Case Studies

2048

Go (圍棋)

MCTS, AlphaGo, CGI

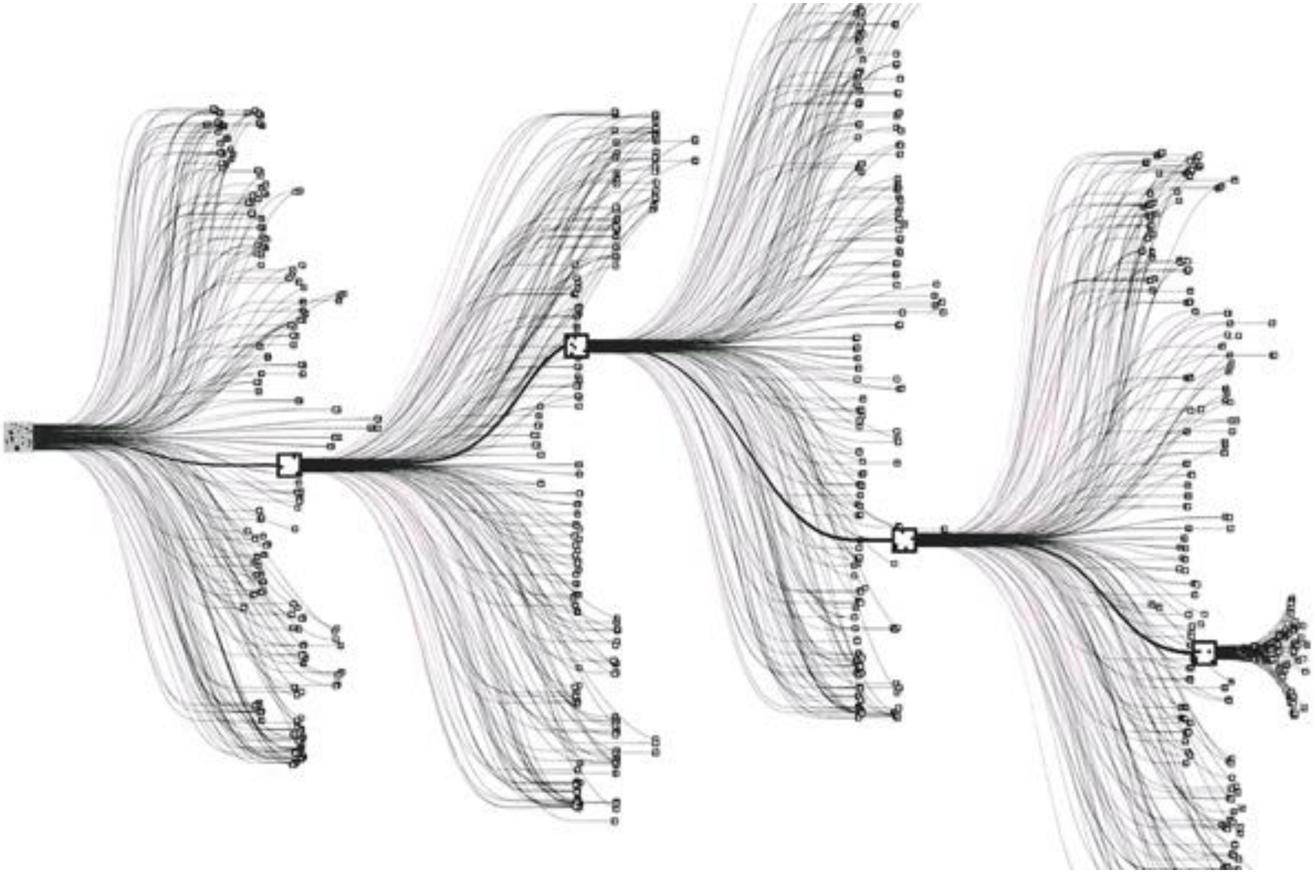
Video Games

Robotics

研究成果摘述 ▶

圍棋(Go) – 最複雜的熱門棋類遊戲

- 遊戲複雜度：約有10^{360}種變化
  - 電腦無法嘗試所有走法

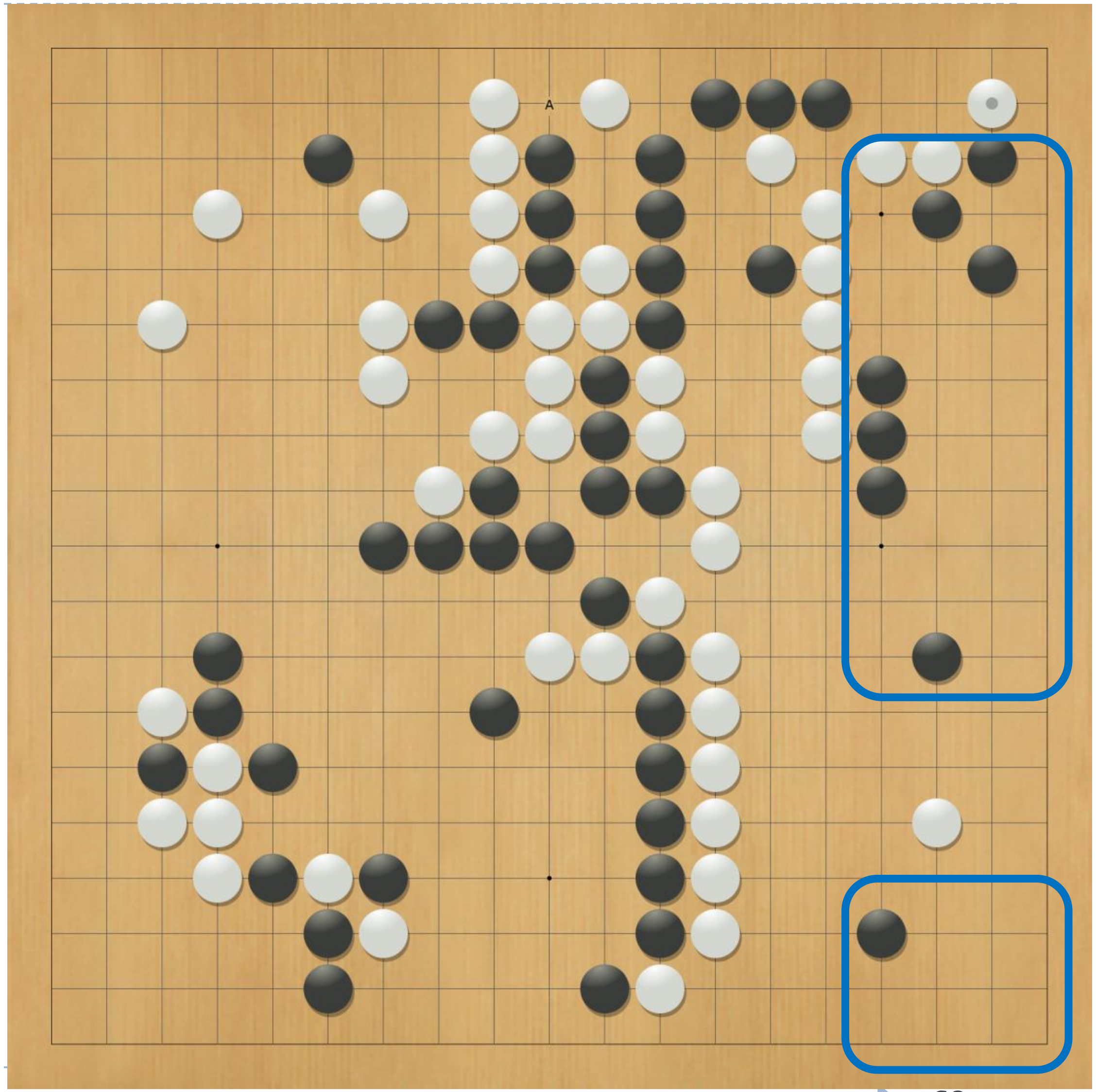


Why not alpha-beta search for Go?

- 圍棋只有兩種子
  - 無法簡單給分評估
- 必須判斷死活
  - 但許多是連動的
  - 如右方黑子死活影響右下黑子
- 但判斷死活等同一複雜搜尋.

因此, 既然都不准

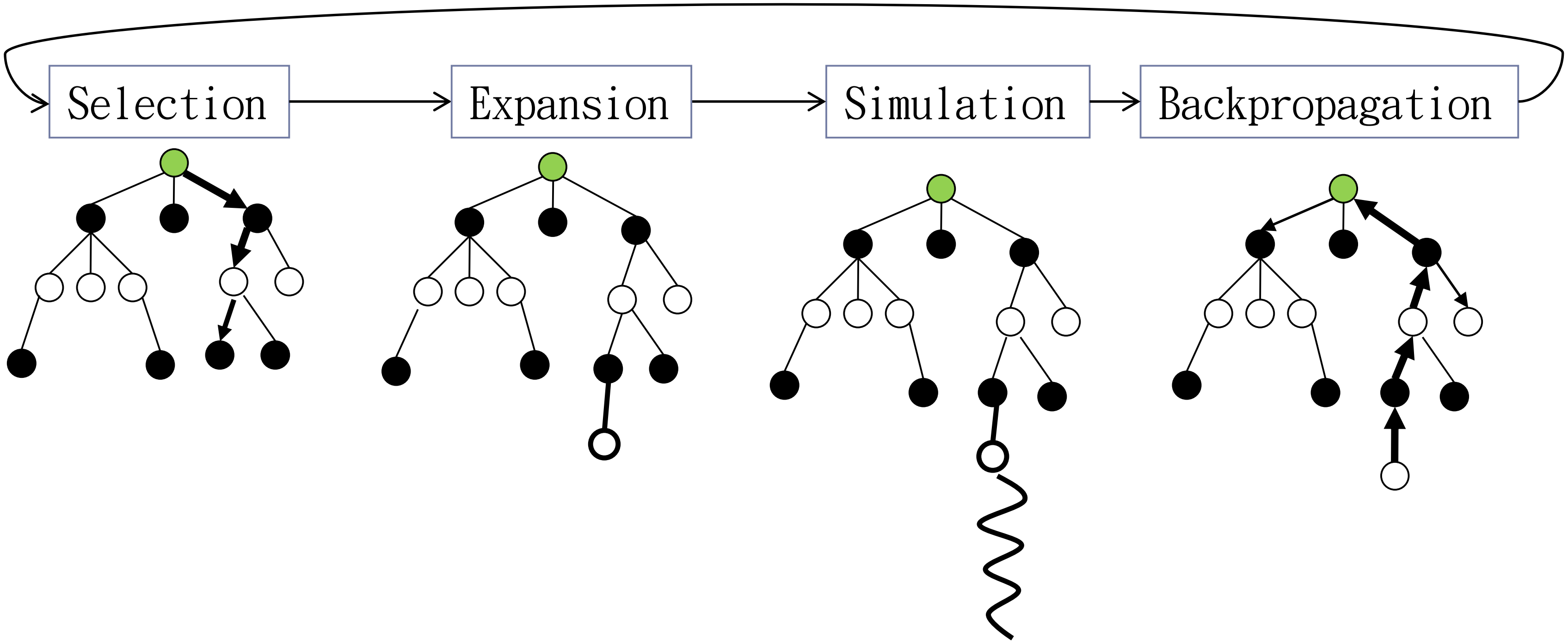
- 何不走到底?
- 用統計勝率來評估



Game 1: AlphaGo vs. 李世石

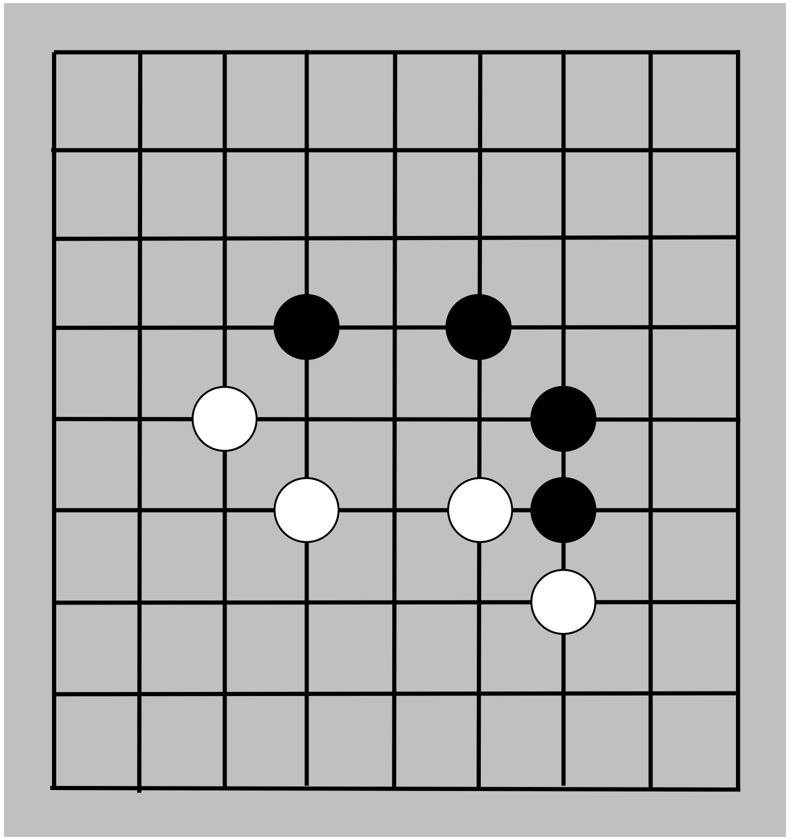
蒙地卡羅樹搜尋

- Monte-Carlo Tree Search (MCTS)
- 一種 Reinforcement learning方法



Rules Overview Through a Game (opening 1)

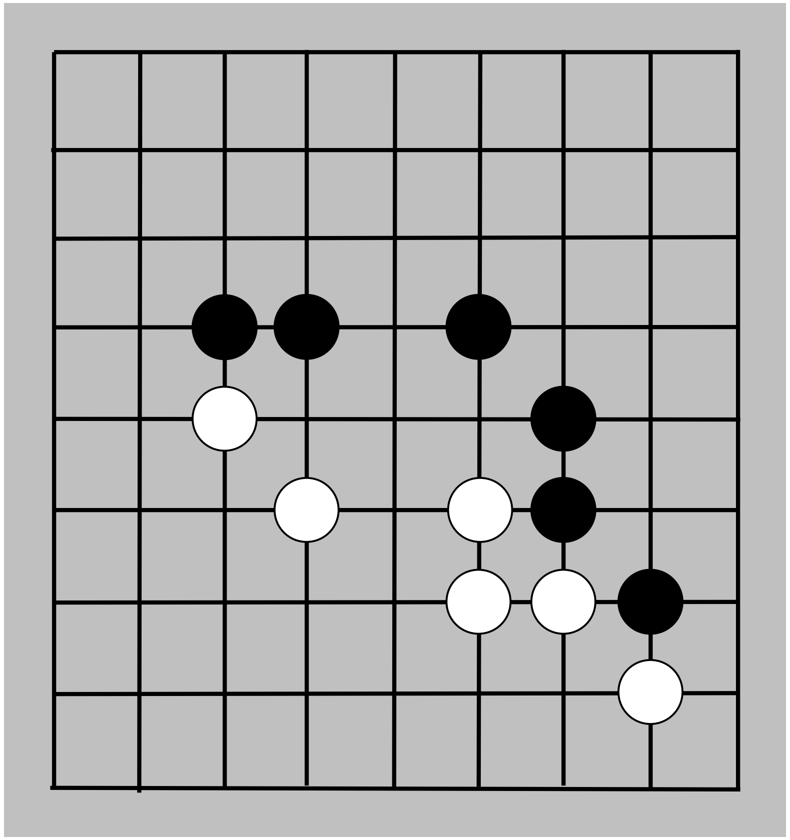
- Black/White move alternately by putting one stone on an intersection of the board.



The example was given by B. Bouzy at CIG'07.

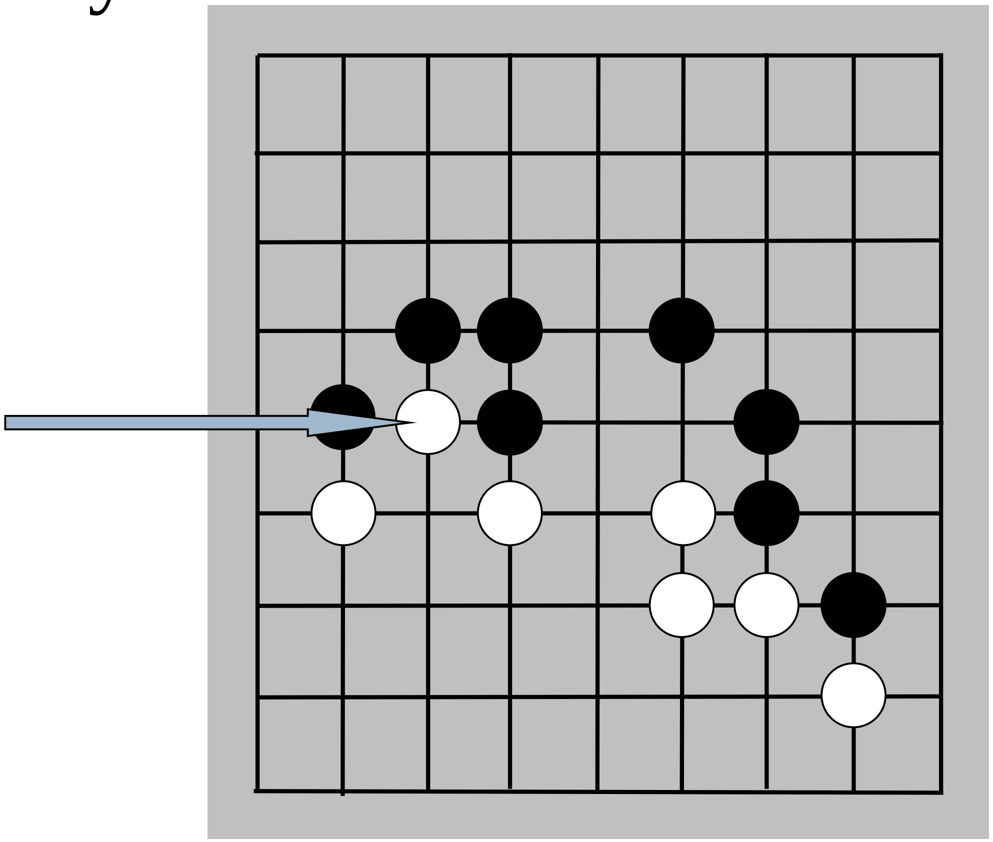
Rules Overview Through a Game (opening 2)

- Black and White aims at surrounding large « zones »



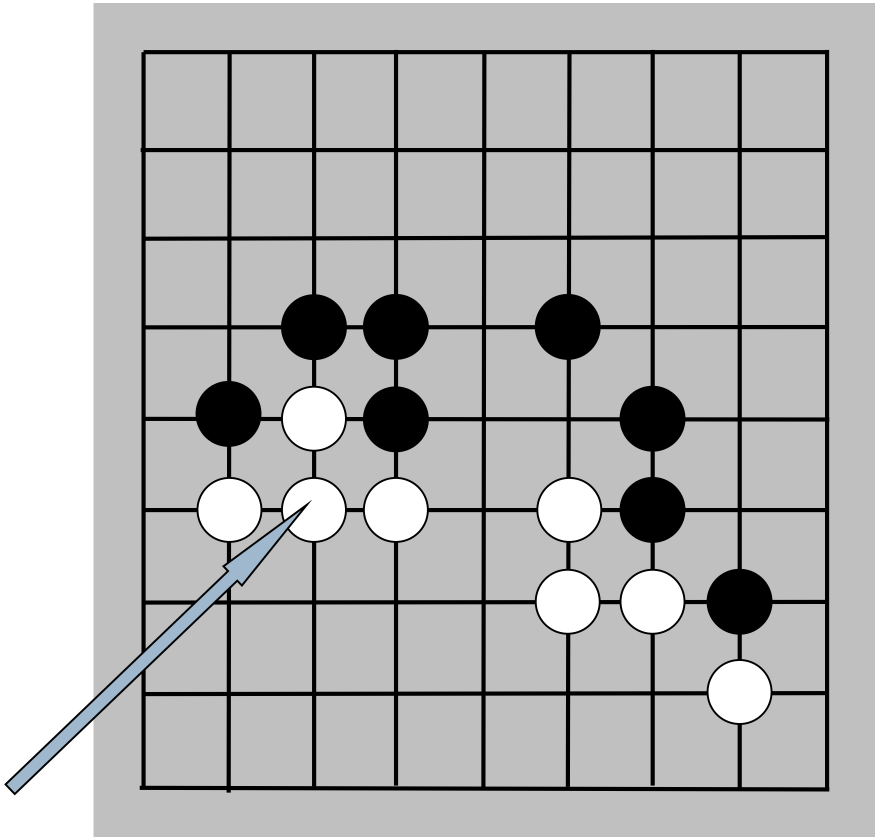
Rules Overview Through a Game (atari 1)

- A white stone is put into « atari » : it has only one liberty left.



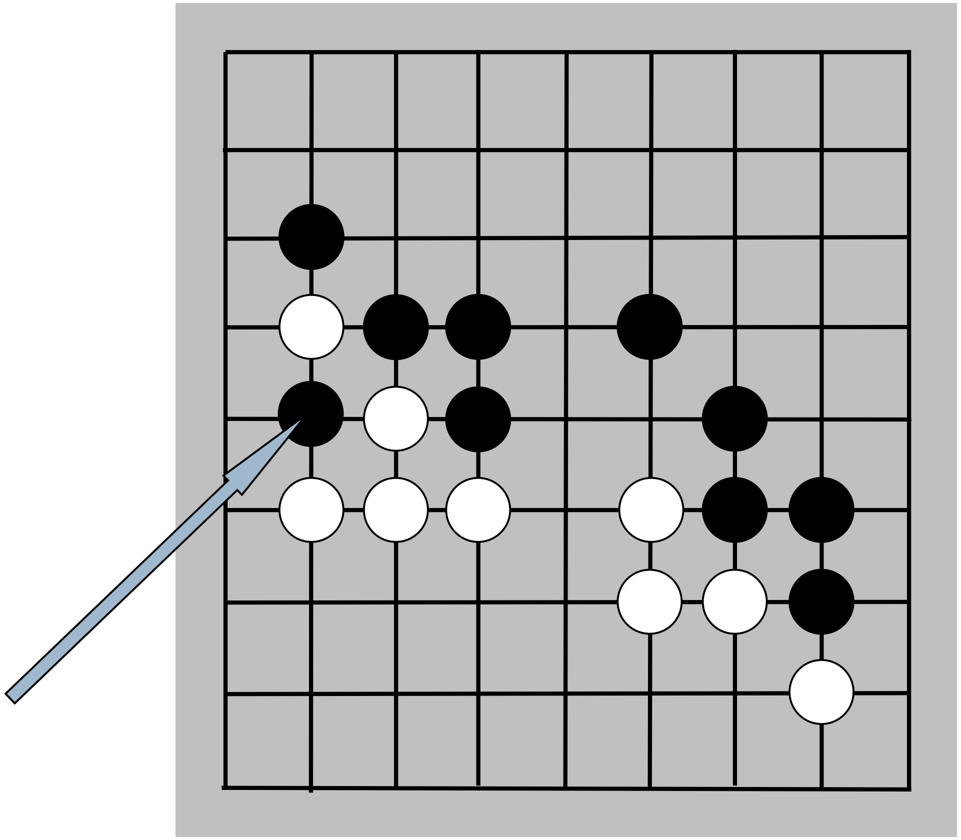
Rules Overview Through a Game (defense)

- White plays to connect the one-liberty stone yielding a four-stone white string with 5 liberties.



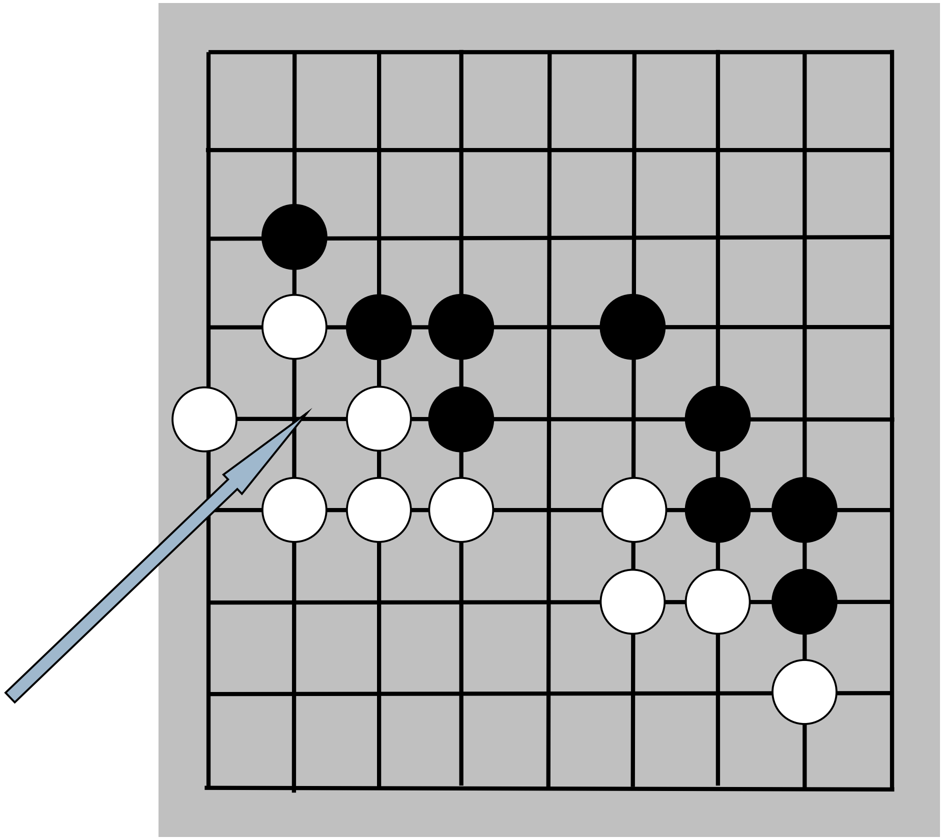
Rules Overview Through a Game (atari 2)

- It is White’s turn. One black stone is atari.



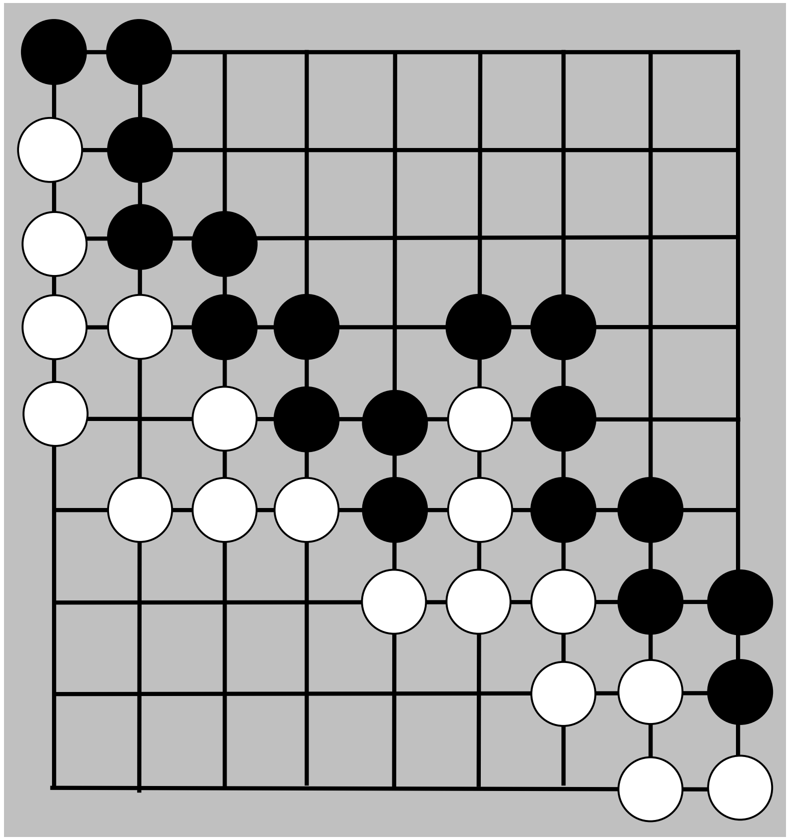
Rules Overview Through a Game (capture 1)

- White plays on the last liberty of the black stone which is removed



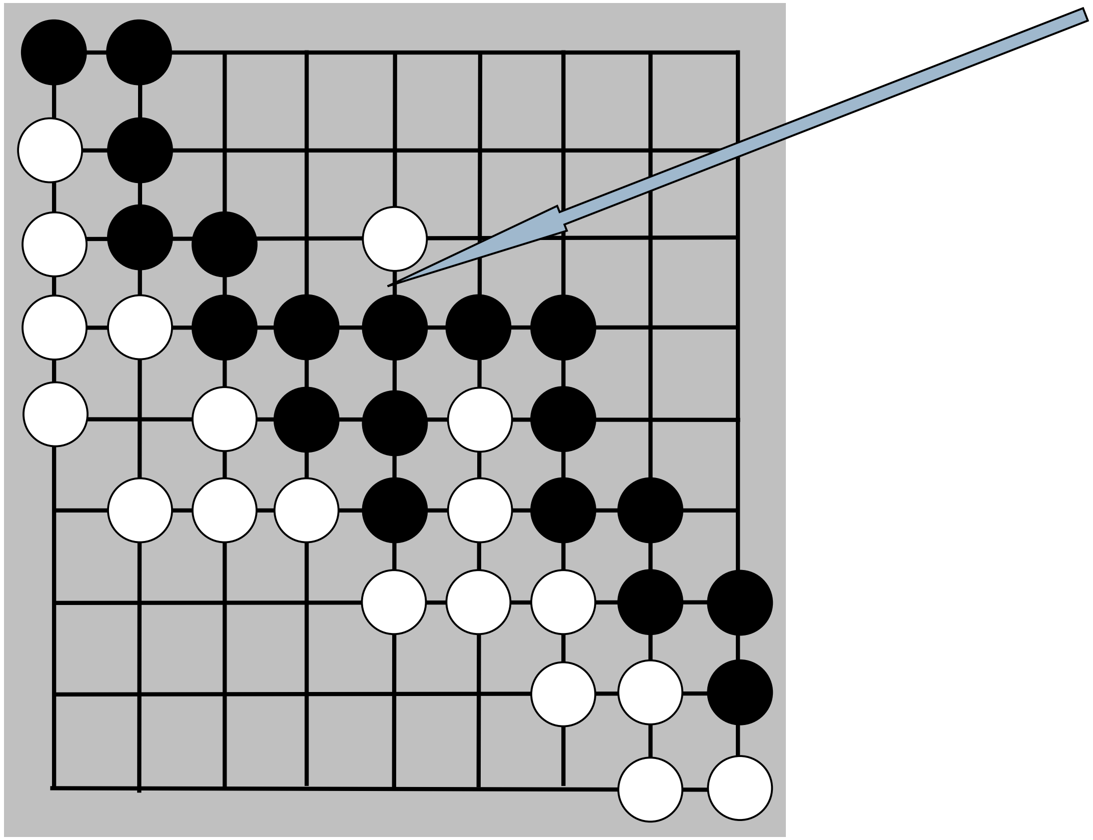
Rules Overview Through a Game (human end of game)

- The game ends when the two players pass. (Experts would stop here)



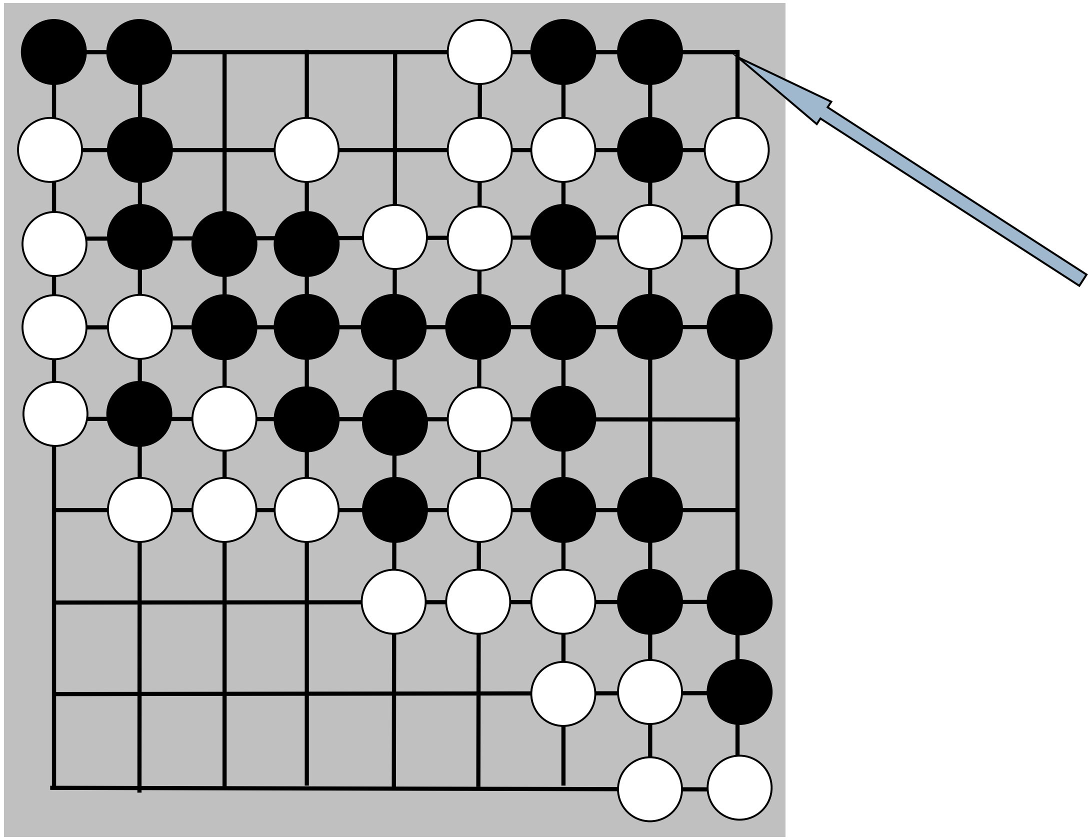
Rules Overview Through a Game (contestation 1)

- White contests the black « territory » by playing inside.



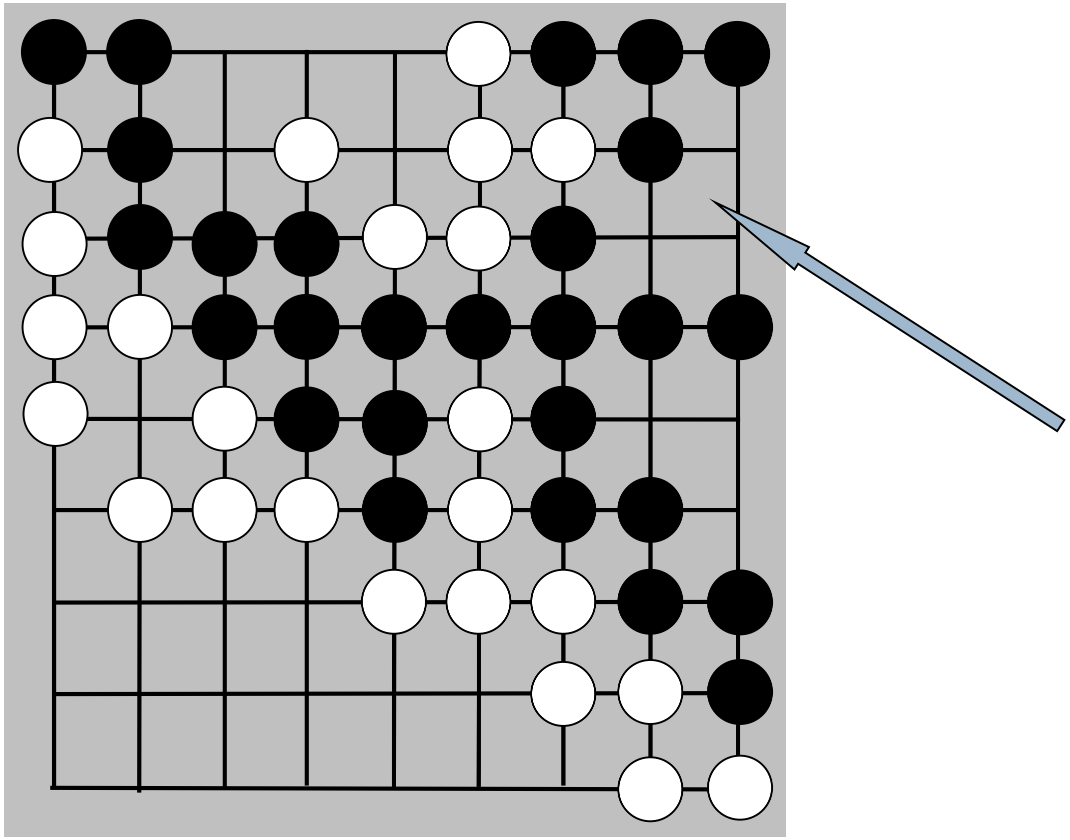
Rules Overview Through a Game (contestation 2)

- White contests black territory, but the 3-stone white string has one liberty left



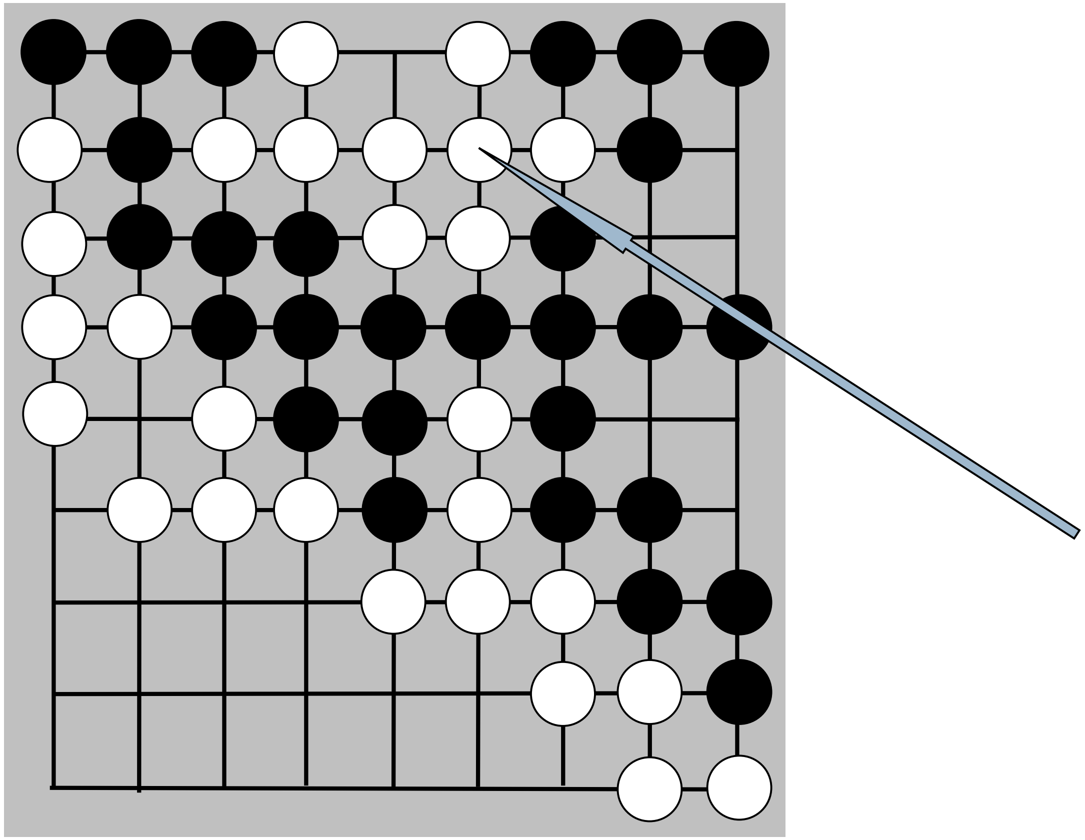
Rules Overview Through a Game (follow up 1)

- Black has captured the 3-stone white string



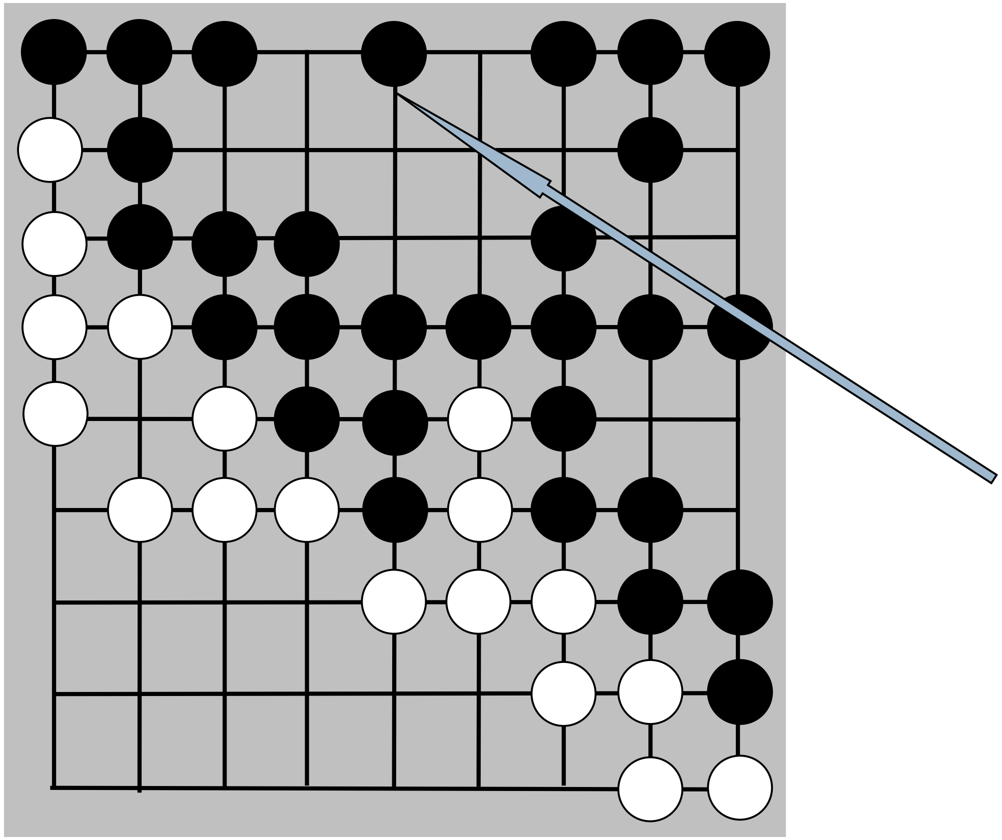
Rules Overview Through a Game (follow up 2)

- White lacks liberties…



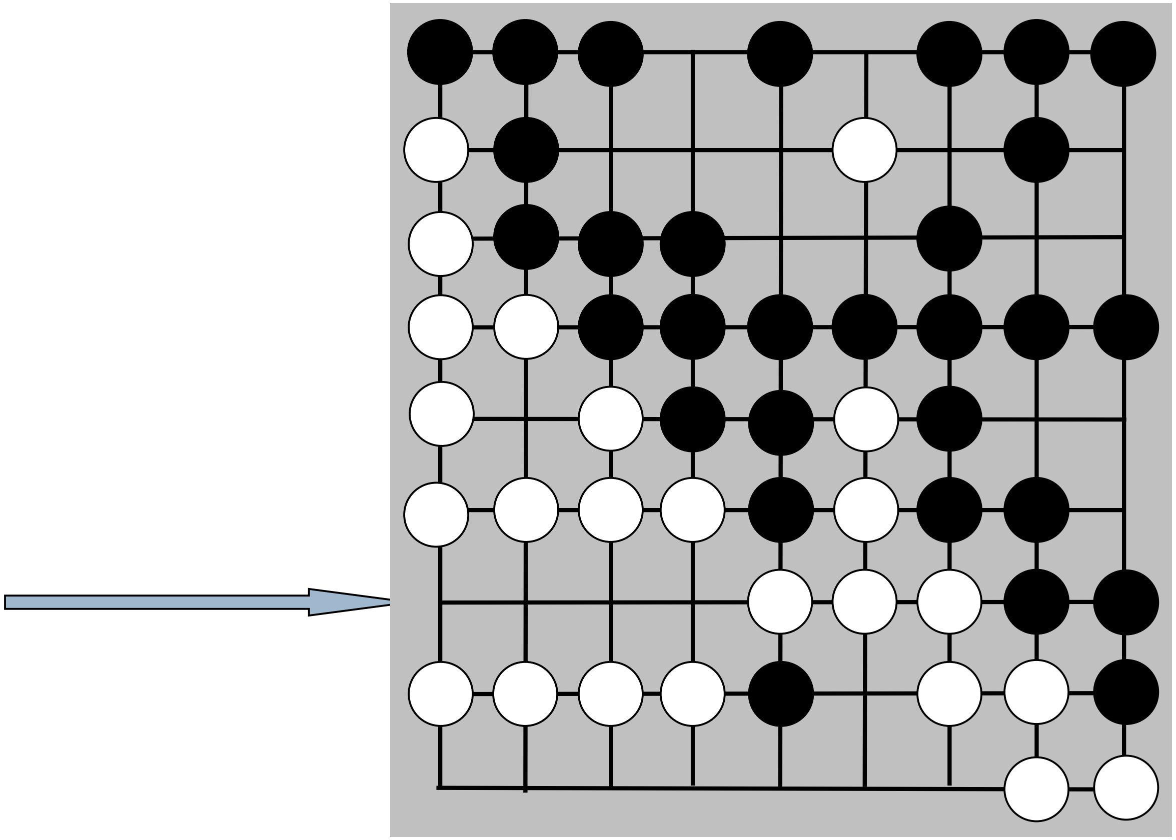
Rules Overview Through a Game (follow up 3)

- Black suppresses the last liberty of the 9-stone string
- Consequently, the white string is removed



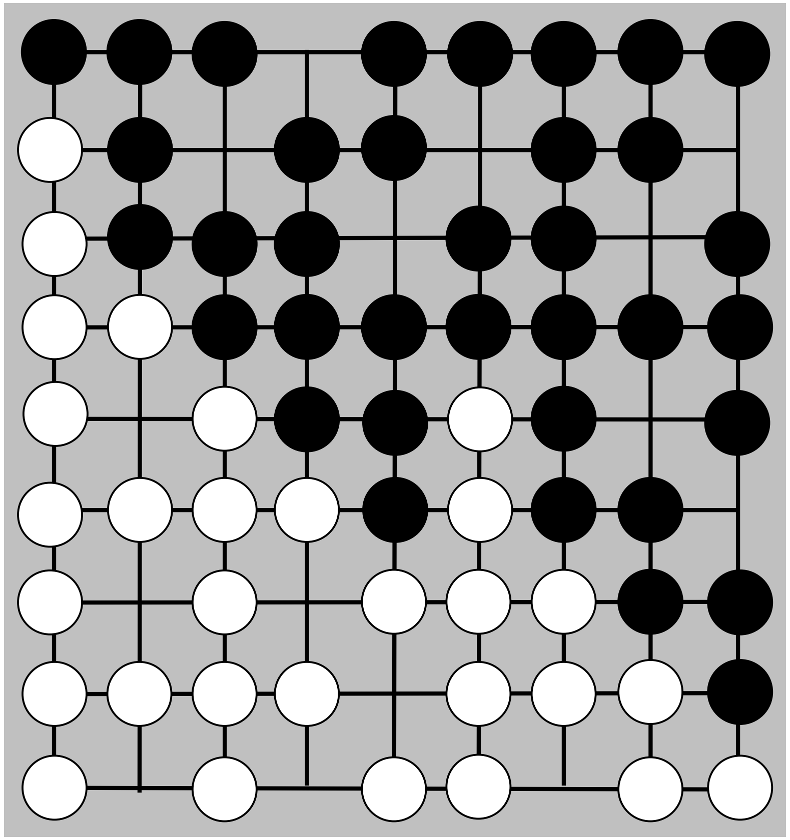
Rules Overview Through a Game (follow up 4)

- Contestation is going on. White has captured four black stones.



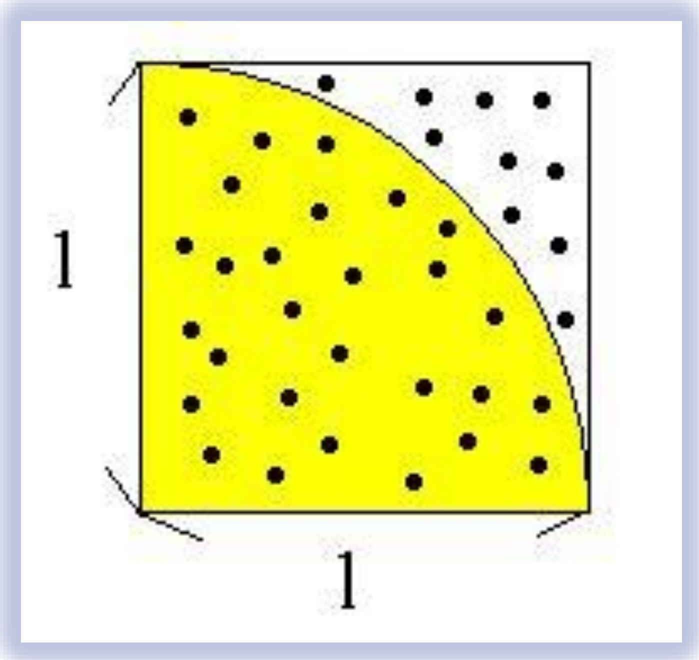
Rules Overview Through a Game (concrete end of game)

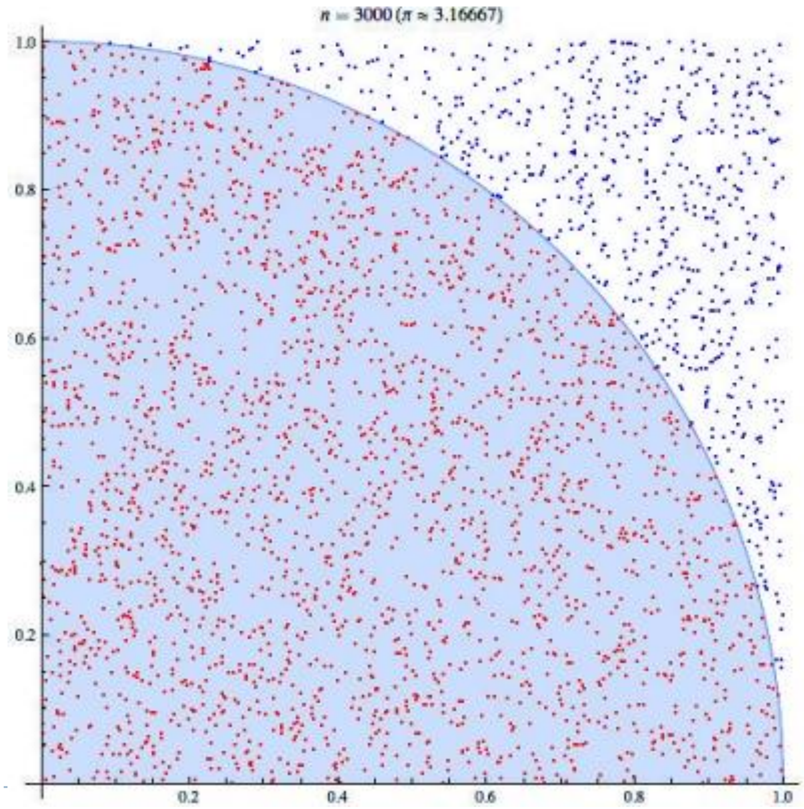
- The board is covered with either stones or « eyes ». Programs know to end.



基本原理

- 利用統計學原理, 來計算.
- 有名的例子: 計算 π.





吃角子老虎問題(Bandit Problem)

- 10台機器, 有1台可以賺錢, 但不知道哪台
- 問題: 有無限多次, 如何賺最多錢?

探索(Exploration) vs. 開發(Exploitation)

- Example for the exploration vs exploitation dilemma
  - **Exploration**: is a long-term process, with a risky, uncertain outcome.
  - **Exploitation**: by contrast is short-term, with immediate, relatively certain benefits

Deterministic Policy: UCB1

- UCB: Upper Confidence Bounds. [Auer *et al.*, 2002]
- Observed rewards when playing machine *i*: $X_{i,1}$, $X_{i,2}$, ...
- Initialization: Play each machine once.
- Loop:
  - Play machine *j* that maximizes,

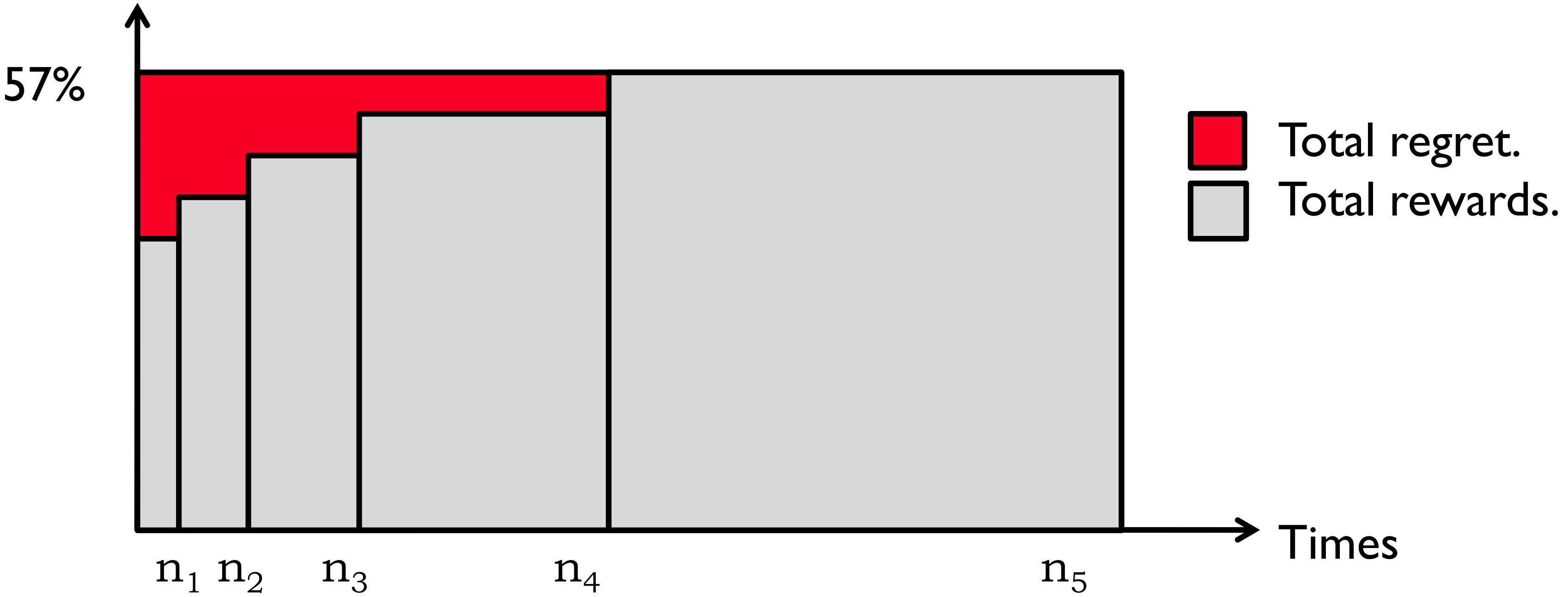
    $$\bar{X}_j + \sqrt{\frac{2\log n}{T_j(n)}}$$

    where *n* is the overall number of plays done so far,

    $$\bar{X}_{i,s} = \frac{1}{s}\sum_{j=1}^{s} X_{i,j} \quad , \quad \bar{X}_i = \bar{X}_{i,T_i(n)} \, ,$$
- Key:
  - Ensure optimal machine is played exponentially more often than any other machine.

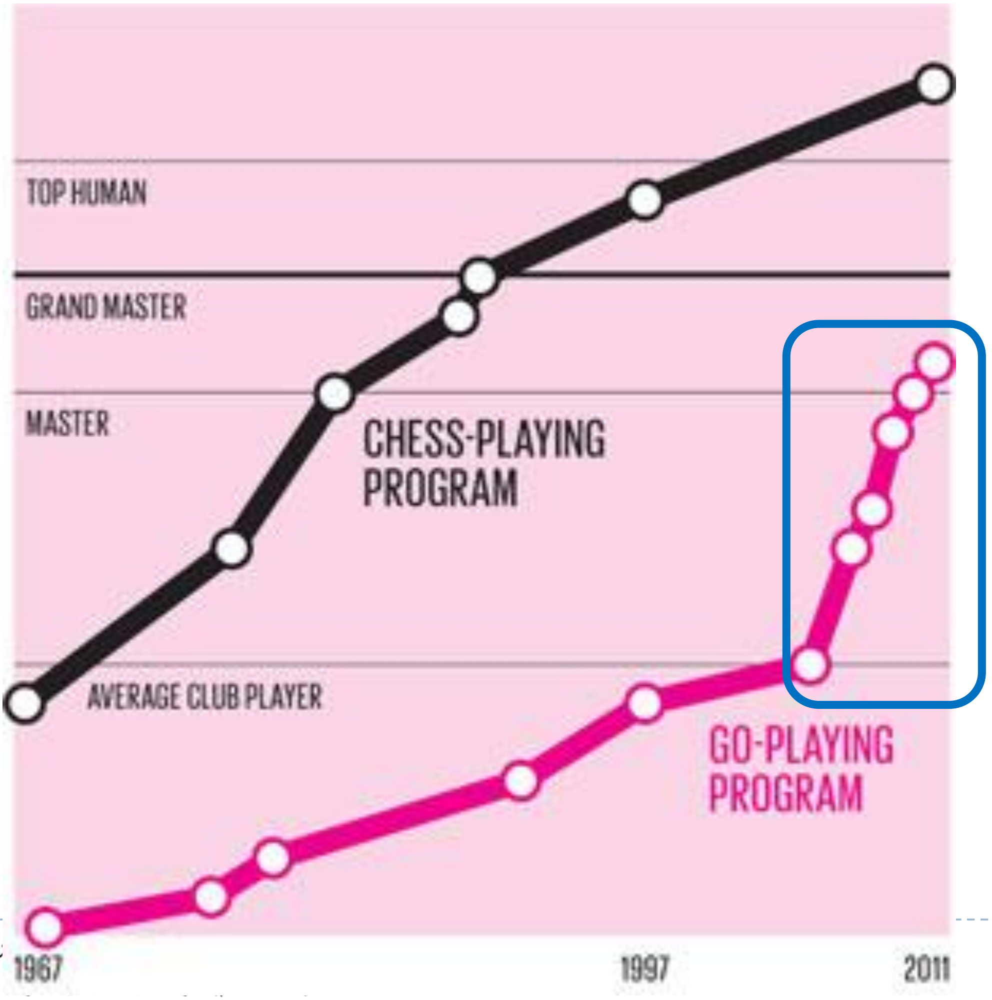
Cumulative Regret

- Assume Machines M_1, M_2, M_3, M_4, M_5
  - Win rates: 37%, 42%, 47%, 52%, 57%
  - Trial numbers: n_1, n_2, n_3, n_4, n_5.



Strength of Go Program after MCTS

- [Schaeffer et al., 2014]



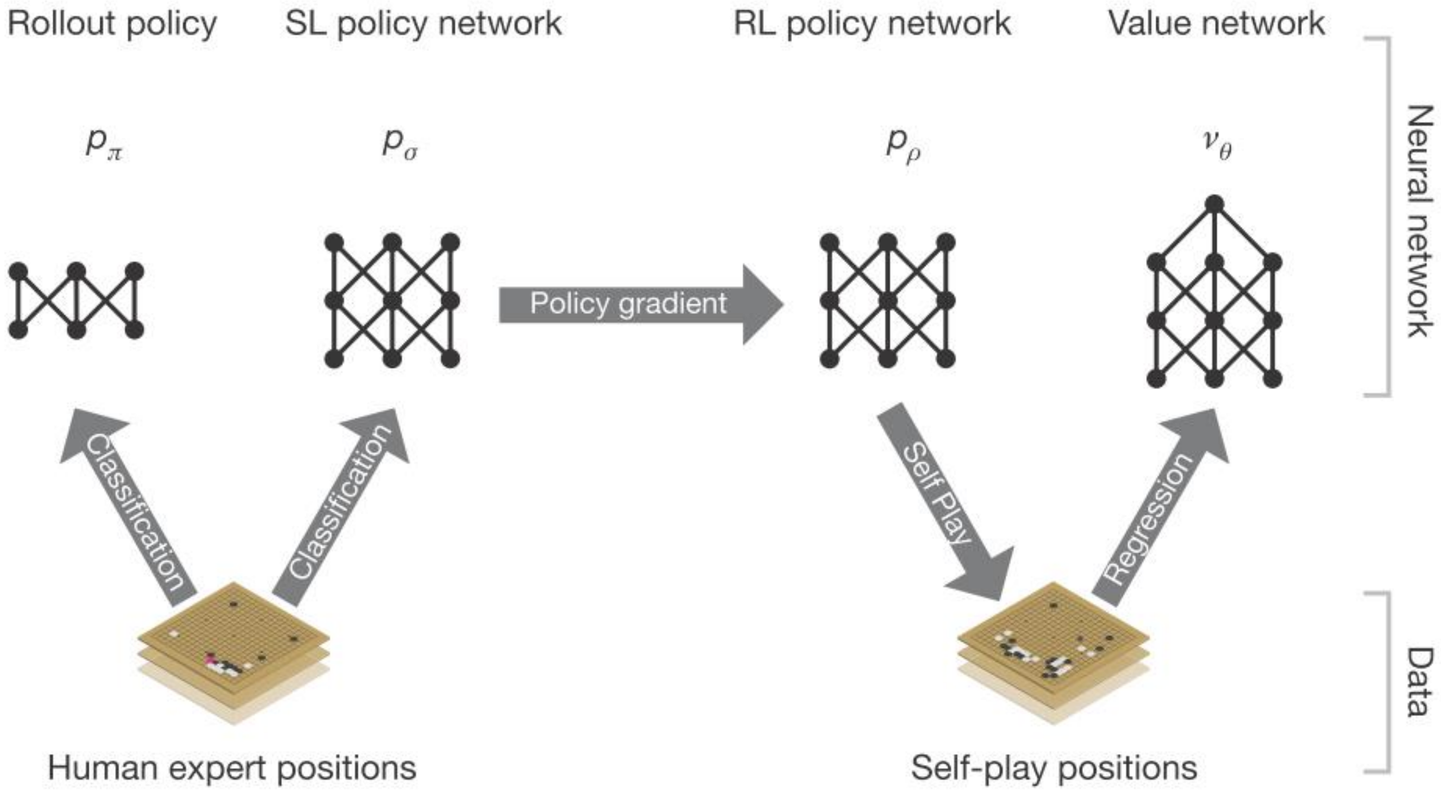
MCTS 出現後
圍棋棋力的成長

AlphaGo

AlphaGo的技術特點

- 採用Monte-Carlo Tree Search (MCTS) → RL
  - 可以自主搜尋, 找尋最佳的下法, 避開陷阱.
- 利用 DCNN 辨識棋型, 學習高手的著手策略,
  - 找出最可能的下法, 並專注在這些下法. → DL
- 以DCNN, 設計 “reinforcement learning (RL) network” (強化式學習) → DRL (Policy Gradient).
  - 自主學習：利用自我對打, 學習更好更新的下法.
- 以 DCNN, 設計 “value network” (價值網路)
  - 學習盤面局勢之優劣 →DL

Policy Network and Value Network

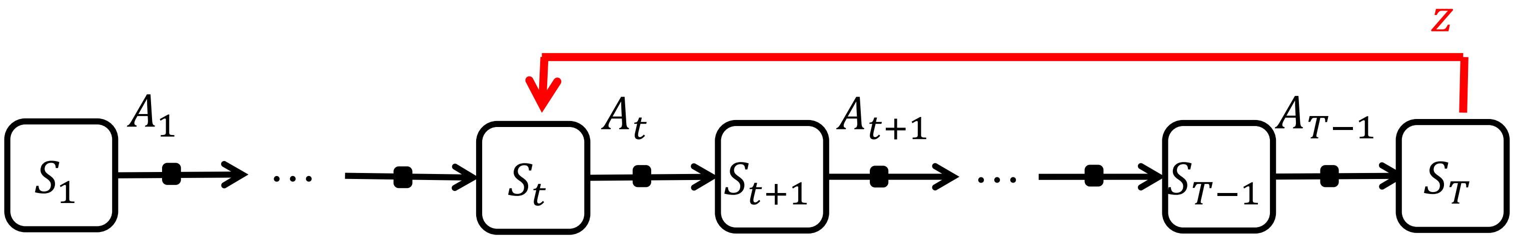


RL Policy Network: AlphaGo

- Use stochastic policy gradient ascent to maximize the likelihood of the human move a selected in state s

$$\Delta\theta = \alpha \nabla_\theta \log \pi_\theta(s_t, a_t) \cdot z$$

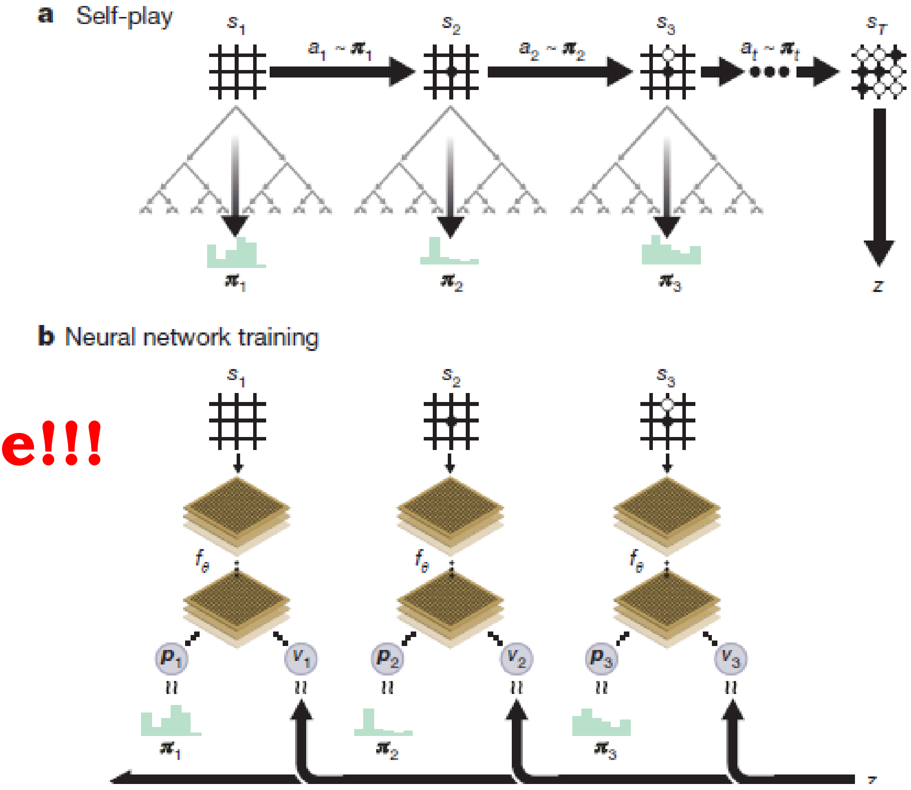
  - θ: network parameter.
  - α: learning rate
  - z: the value of the episode
    - win/loss (1/-1) of the game



AlphaGo Zero

- 採用Monte-Carlo Tree Search (MCTS) → RL
  - 可以自主搜尋, 找尋最佳的下法, 避開陷阱.
- Combine “value/policy network” →DRL

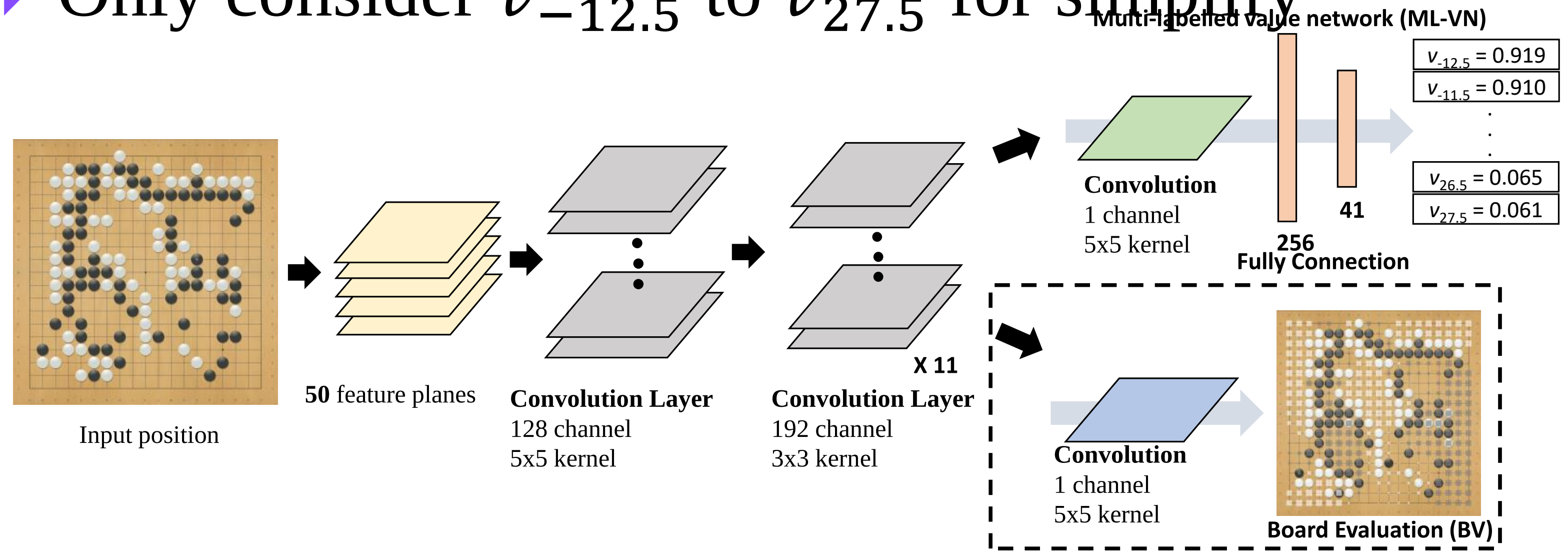
Learn from Zero Knowledge!!!



CGI Go Intelligence
Multi-labelled Value Network
(by Our Lab)

Multi-labelled (ML) Value Network

- Includes all value outputs v_k for k-komi games (貼k目)
  - The full set of value outputs can be $v_{-361.5}$ to $v_{361.5}$
  - Only consider $v_{-12.5}$ to $v_{27.5}$ for simplify



Multi-labelled value network (ML-VN)

Input position

50 feature planes

Convolution Layer
128 channel
5x5 kernel

X 11

Convolution Layer
192 channel
3x3 kernel

Convolution
1 channel
5x5 kernel

256

41

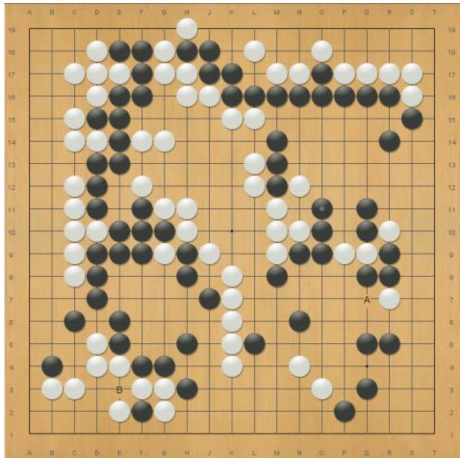
Fully Connection

$v_{-12.5}$ = 0.919
$v_{-11.5}$ = 0.910
.
.
.
$v_{26.5}$ = 0.065
$v_{27.5}$ = 0.061

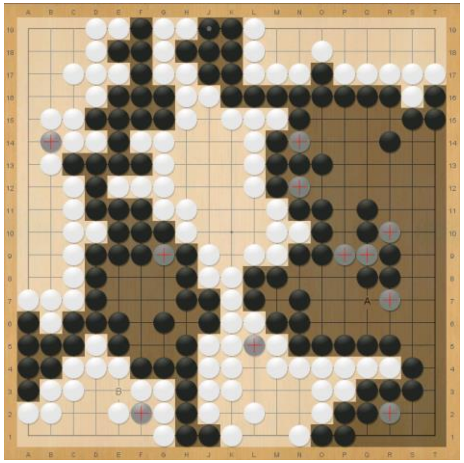
Convolution
1 channel
5x5 kernel

Board Evaluation (BV)

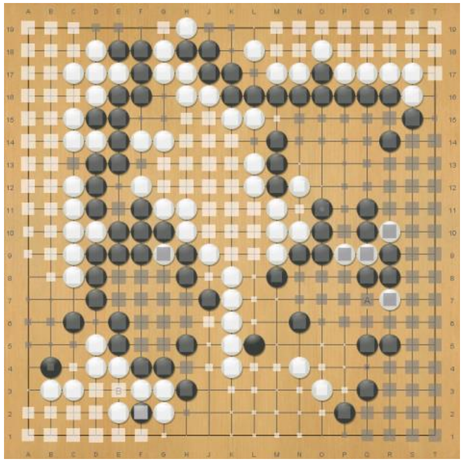
Label ML Value Network



Current positions (Training data)



Final positions



Output of Board Evaluation

k (komi)	Label on v_k	v_k (win rate)
0.5	1	0.678897
1.5	1	0.599618
2.5	1	0.599108
3.5	-1	0.512413
4.5	-1	0.511263
5.5	-1	0.423886
6.5	-1	0.423626
7.5	-1	0.339738
8.5	-1	0.339353

Strengths of Different Value Networks

- Setting
  - One GPU and six CPU cores for each program.
  - 500 games are played with 1 second each move.
- Results (for Komi 7.5 only)
  - BV-ML-VN, BV-VN and ML-VN outperform VN only
  - BV-ML-VN performs the best

Network	VN	ML-VN	BV-VN	BV-ML-VN
VN	-	39.60% (±4.29%)	39.40% (±4.29%)	32.40% (±4.11%)
ML-VN	60.40% (±4.29%)	-	49.20% (±4.39%)	47.20% (±4.38%)
BV-VN	66.60% (±4.29%)	50.80% (±4.39%)	-	47.20% (±4.38%)
BV-ML-VN	**67.60% (±4.11%)**	**52.80% (±4.38%)**	**52.80% (±4.38%)**	-

案例研究
Case Studies

2048

Go (圍棋)

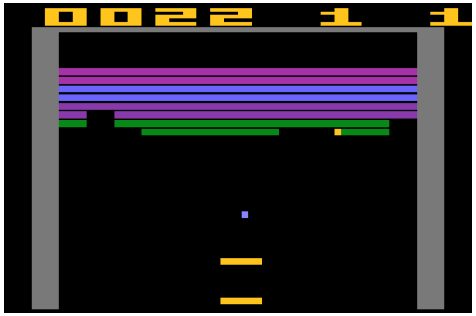
Video Games

Robotics

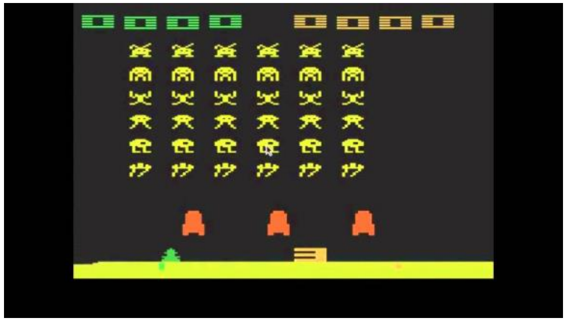
Deep Q-Learning for Atari 2600 Games

- Learn to play Atari games from video only (without knowing the game a priori)

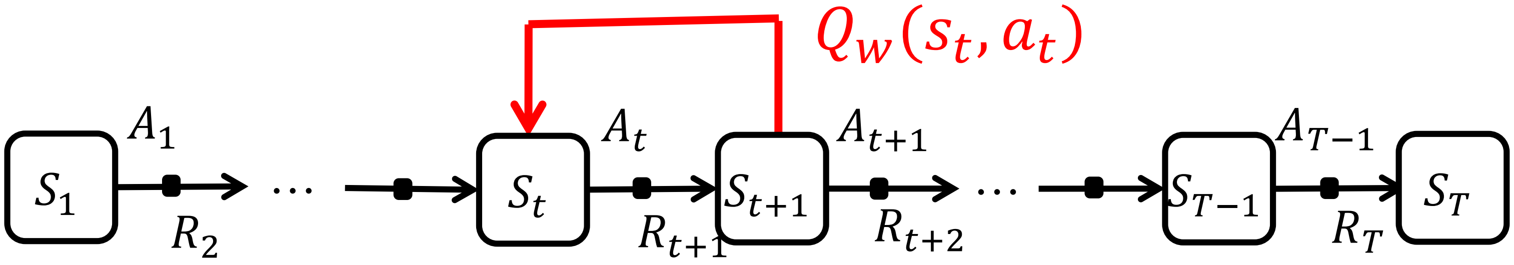
- Atari 2600



- Breakout



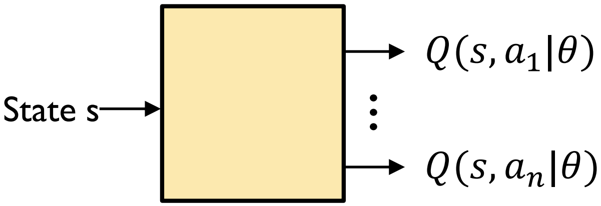
- Space Invaders



Replay Buffer: a set of (s_t, a_t, r_t, s_{t+1})

Deep Q Network (DQN)

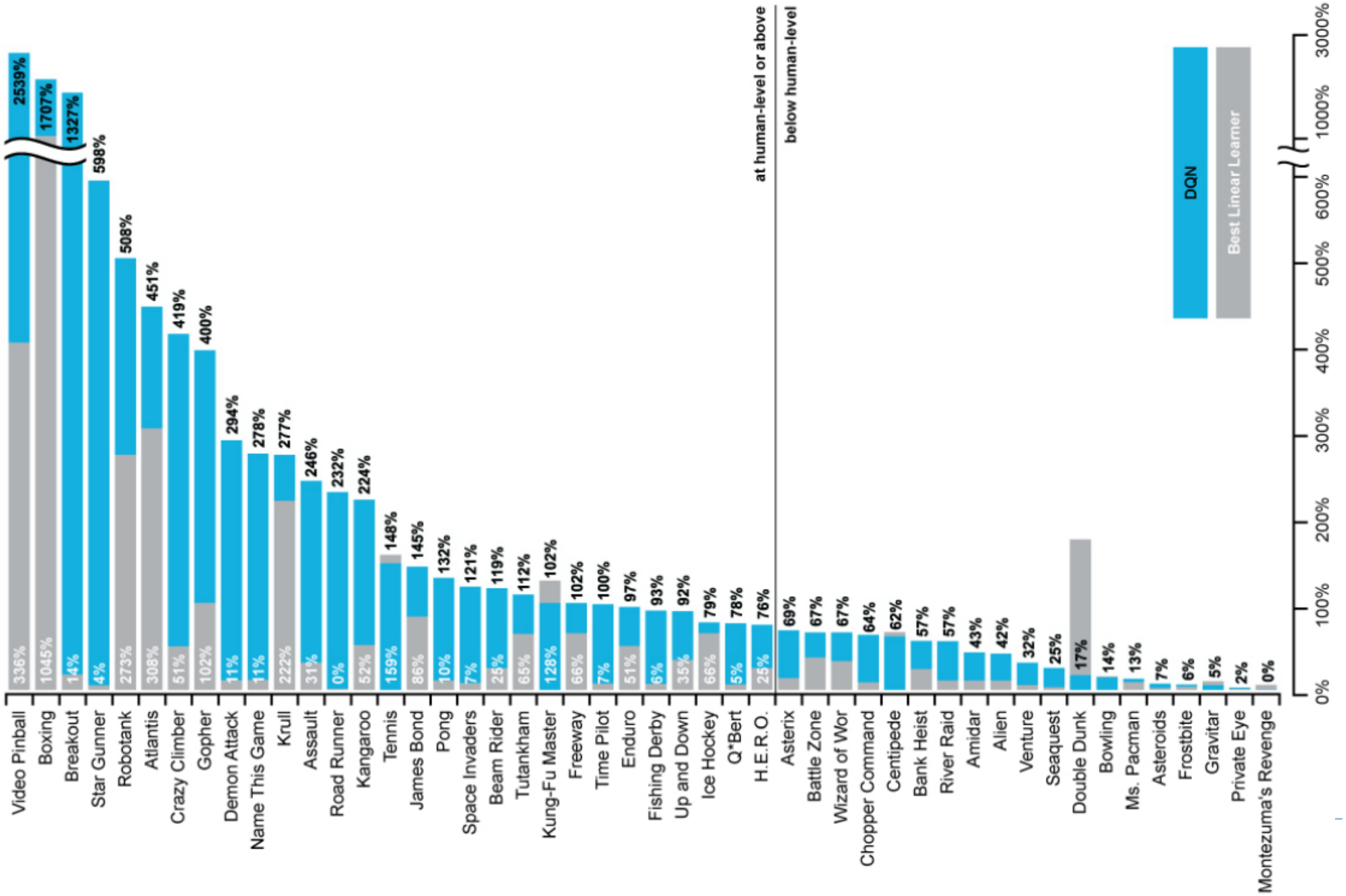
- Single deep network estimates the action value function of each discrete action
  - Action Value: $Q(s_t, a_t|\theta)$
  - Select action: $\arg\max_{a\prime} Q(s_t, a'|\theta)$
- A kind of TD learning (TD(0))
  - Target Q value: (TD Target)
    - $Y_t^Q = r_{t+1} + \gamma \max_{a\prime} Q(s_{t+1}, a'|\theta)$
  - Loss Function: (Square of TD error)
    - $L_Q(s_t, a_t|\theta) = \left(Y_t^Q - Q(s_t, a_t|\theta)\right)^2$
  - Gradient descent:
    - $\nabla_\theta L_Q(s_t, a_t|\theta) = \left(Y_t^Q - Q(s_t, a_t|\theta)\right)\nabla_\theta Q(s_t, a_t|\theta)$



- Implementation issues:
  - Use experience replays.
  - Use target network θ^- and behavior network θ. (Sync every N=10000)

Performance of Deep Q-Learning

- Left (stronger than human)



案例研究
Case Studies

2048

Go (圍棋)

Video Games

Robotics

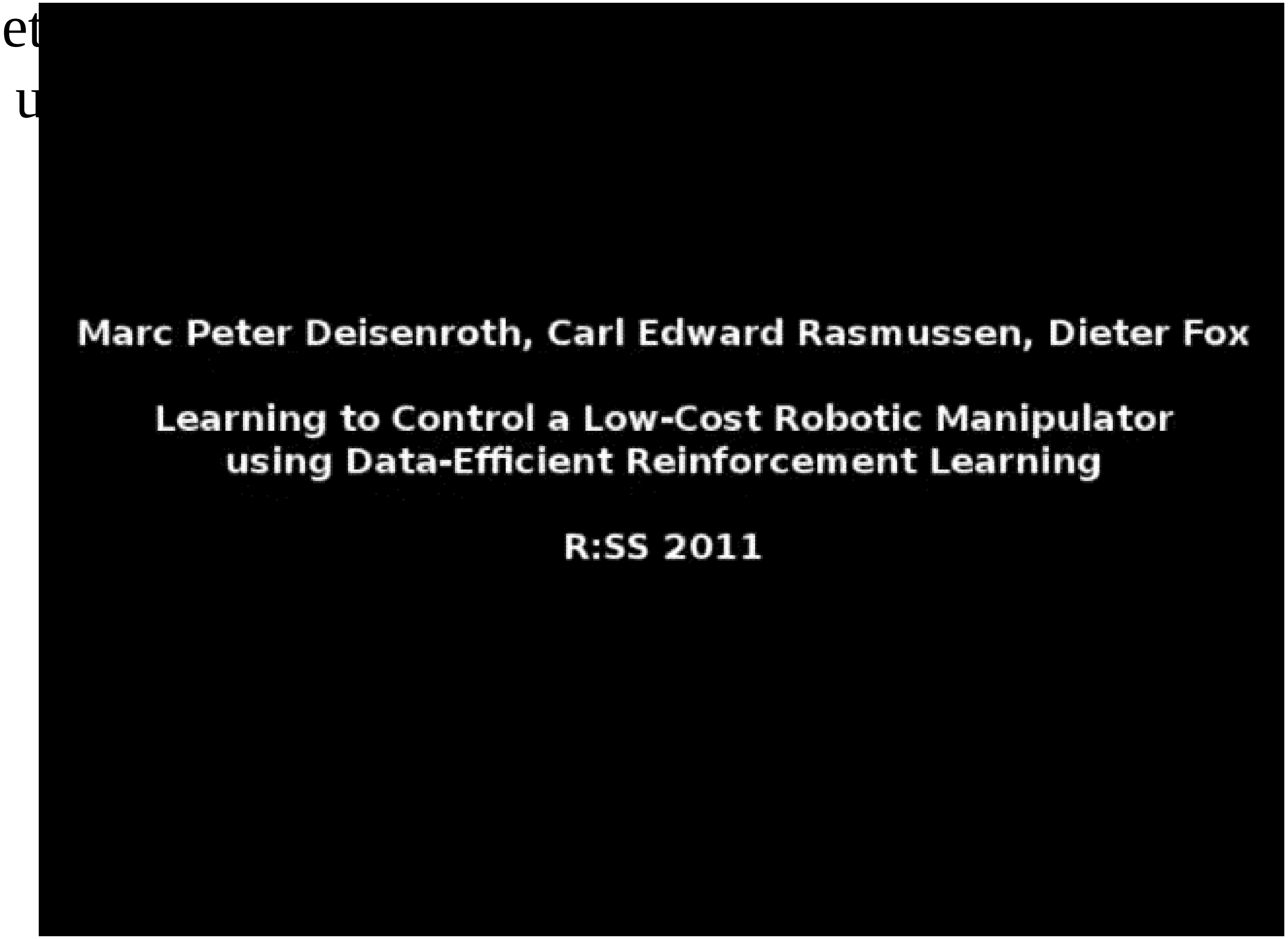
Applications

- Autonomous Driving
- Drone
  - E.g., Precision Landing, Object Tracking
- Robotics
  - E.g., Random Bin Picking (RBP; 隨機工件夾取),
- Learning professional skills
  - Automatic shoveling (自動鏟花)

Robotics Demo (I)

[Deisenroth et

Manipulator u

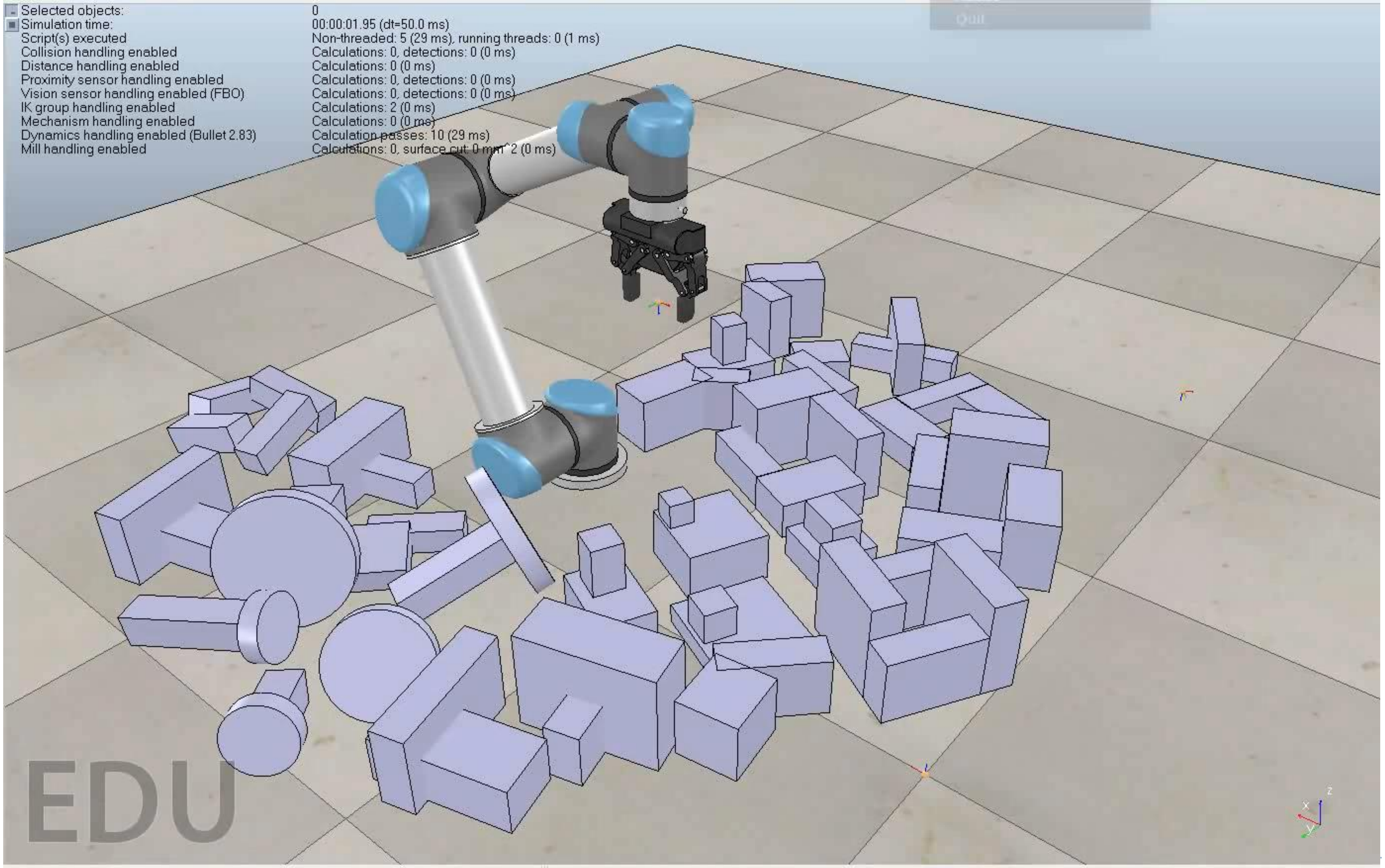


Marc Peter Deisenroth, Carl Edward Rasmussen, Dieter Fox

Learning to Control a Low-Cost Robotic Manipulator
using Data-Efficient Reinforcement Learning

R:SS 2011

Robotics Demo (II) -- RBP



```
Selected objects:                              0
Simulation time:                               00:00:01.95 (dt=50.0 ms)
Script(s) executed                             Non-threaded: 5 (29 ms), running threads: 0 (1 ms)
Collision handling enabled                     Calculations: 0, detections: 0 (0 ms)
Distance handling enabled                      Calculations: 0 (0 ms)
Proximity sensor handling enabled              Calculations: 0, detections: 0 (0 ms)
Vision sensor handling enabled (FBO)           Calculations: 0, detections: 0 (0 ms)
IK group handling enabled                      Calculations: 2 (0 ms)
Mechanism handling enabled                     Calculations: 0 (0 ms)
Dynamics handling enabled (Bullet 2.83)        Calculation passes: 10 (29 ms)
Mill handling enabled                          Calculations: 0, surface cut: 0 mm^2 (0 ms)
```

Rule-based vs. Learning-based

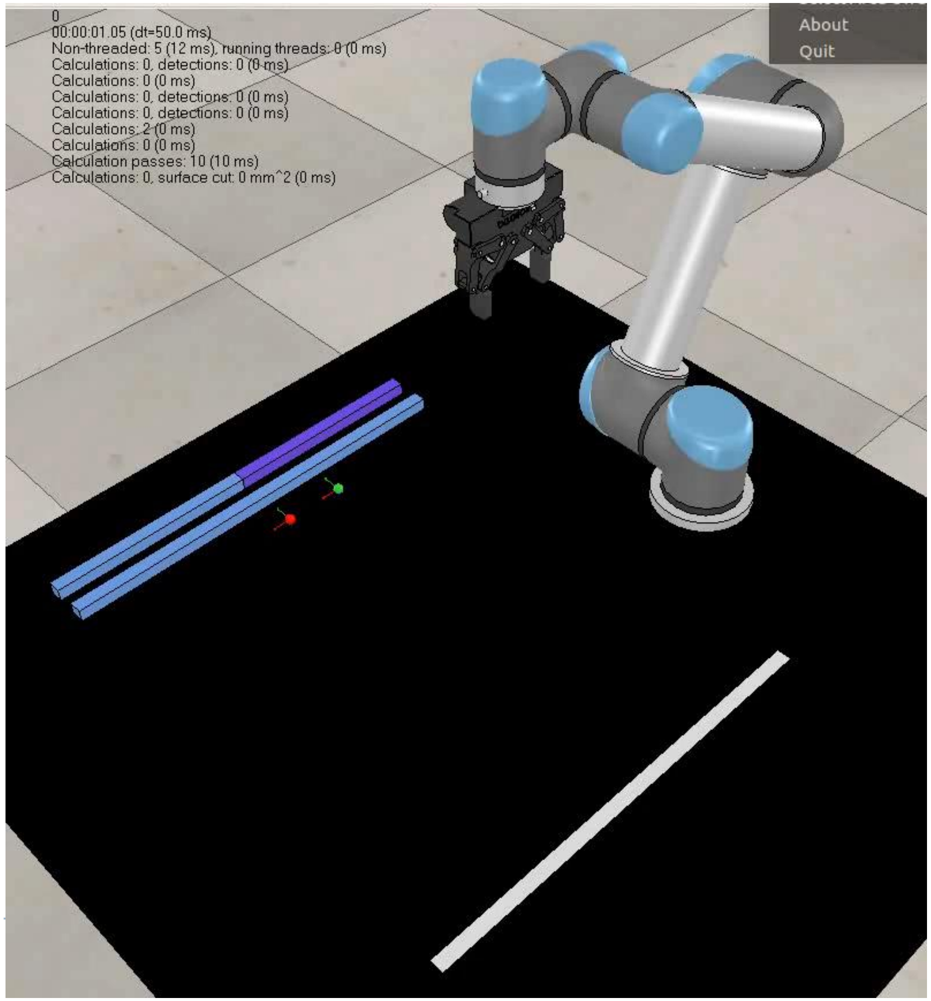
- Rule-based
  - For most traditional industrial robotics.
  - Conditions: environment needs to be well controlled.
  - Drawbacks:
    - Low flexibility.
    - Time required for new jobs.
  - Advantages:
    - Accurate under controlled environment.
- Learning-based (RL/DRL)
  - For most modern service robotics, drones, autonomous driving.
  - Conditions: environment does not need to be well controlled.
  - Goal: Offer high flexibility.

Robotics and Games

Learning Professional Skills

- Automatic shoveling
- Pottery/Clay Molding

A simple demo using RBP



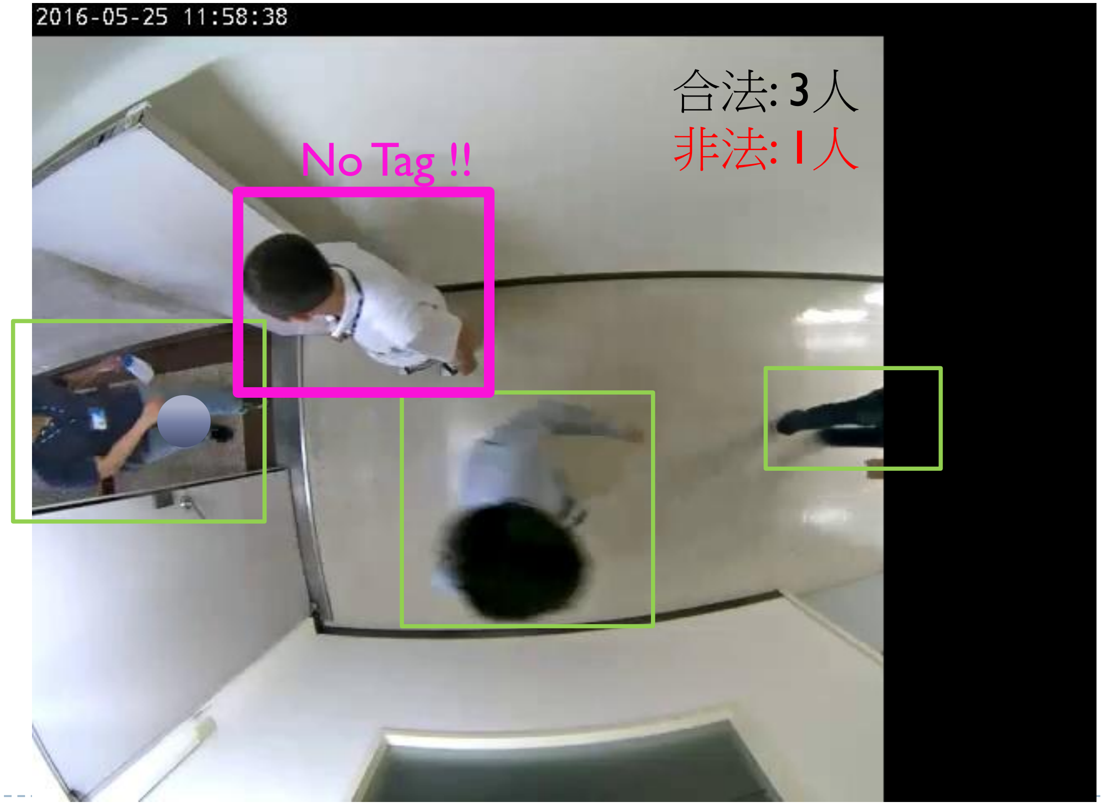
```
0
00:00:01.05 (dt=50.0 ms)
Non-threaded: 5 (12 ms), running threads: 0 (0 ms)
Calculations: 0, detections: 0 (0 ms)
Calculations: 0 (0 ms)
Calculations: 0, detections: 0 (0 ms)
Calculations: 0, detections: 0 (0 ms)
Calculations: 2 (0 ms)
Calculations: 0 (0 ms)
Calculation passes: 10 (10 ms)
Calculations: 0, surface cut: 0 mm^2 (0 ms)
```

Automatic Shoveling

- 機器難取代！"鏟花師"牽動機械業兆元產值 世界翻轉中 20170813

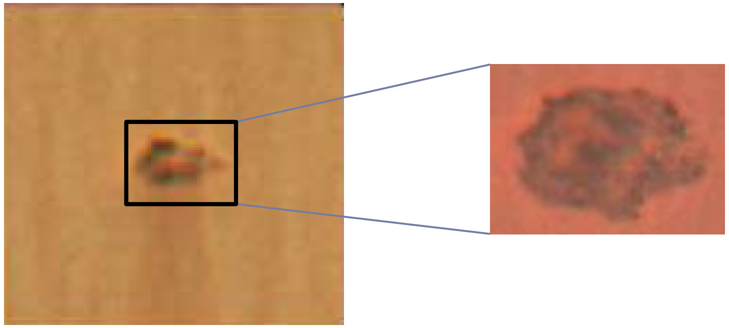
Object Tracking

- Monitoring in factory
- Fusing sensor (like RFID) and visual data



Defect Detection (AOI-like)

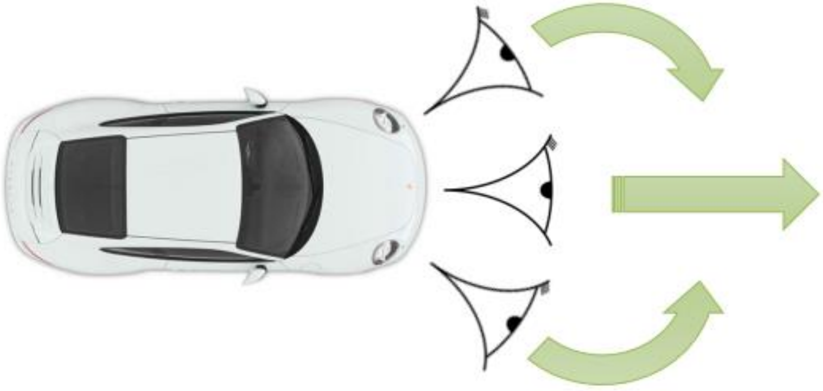
- Room In/Out
  - Reflection of scratches



- Like Gaming

Imitation Learning

- Problem for general machine learning:
  - Training is slow.
  - Cost for training failure (like drone crashes)
- Solution: [Ross et al., 2011]
  - Learn from Demonstration (or Demonstration Cloning)
    - Serve as a bootstrapping process.
    - Some require human helps (like drone)
    - Some don’t (like MCTS, iLQR)
  - Third-Person Imitation Learning



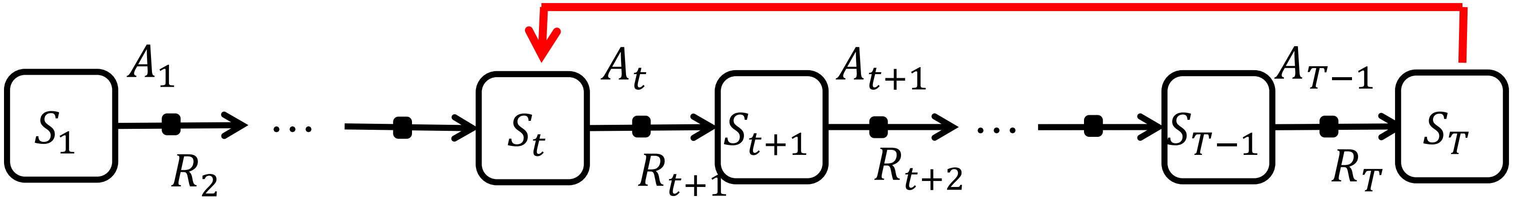
Other Techniques

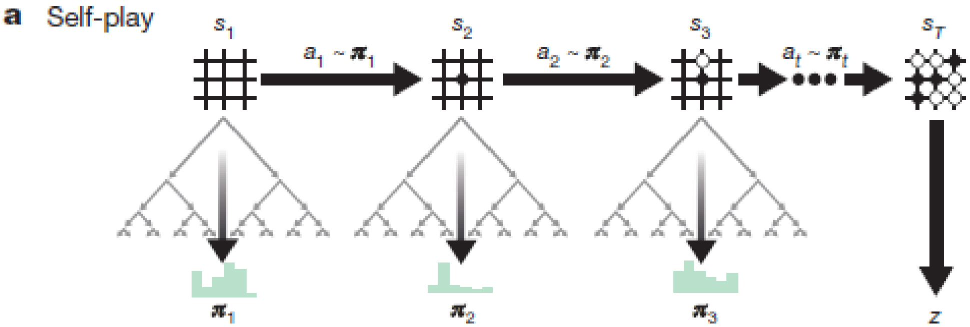
- Curriculum learning
- Transfer Learning
- Behavior Cloning
- Dagger
- GAIL (like GAN)

深度強化式學習應用類型
Application Classification of Deep Reinforcement Learning

Class 1

- Properties:
  - Model is well known or defined
  - Simulator exists.
- Applications: Games, Education, etc.
- Possible Solutions: AlphaZero-like.



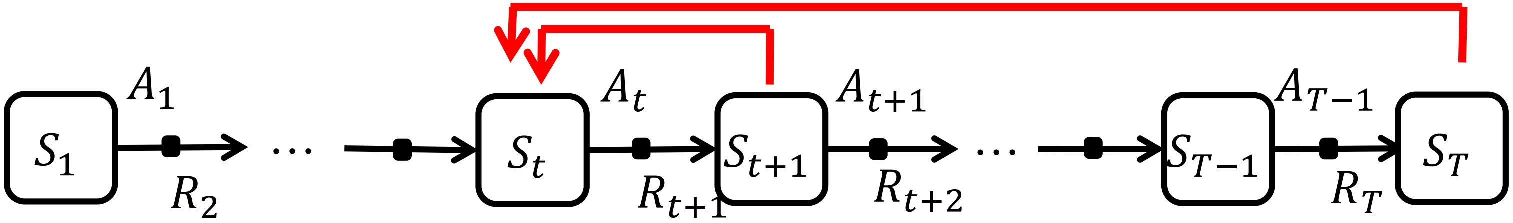


Related Reinforcement Learning Techniques

- TD Learning
- Monte-Carlo Learning
- POMDP
- Monte-Carlo Tree Search (MCTS)
- AlphaZero

Class 2

- Properties:
  - Model is unknown or too complex.
  - Simulator exists.
- Applications: Video Games, Robots with Simulator, etc.
- Possible Solutions: (next page)

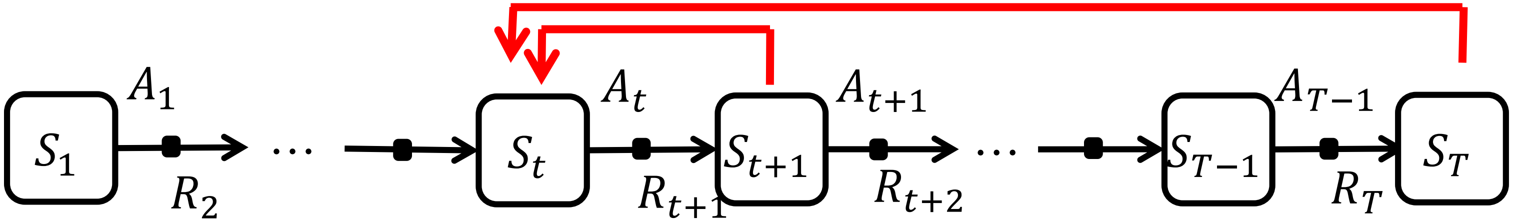


More Related Deep Reinforcement Learning Techniques

- Deep Q Network (DQN)
- Double DQN (DDQN)
- Actor-Critic
- Dueling Network
- Deep Deterministic Policy Gradient (DDPG)
- Asynchronous Advantage Actor-Critic (A3C)
- Trust Region Policy Optimization (TRPO)
- Proximal Policy Optimization (PPO)

Class 3

- Properties:
  - Model is unknown or too complex
  - Simulator does not exist or runs with expensive costs.
    - So, it is hard to produce a large data set.
- Applications: Robots, Drone, Auto-driving, etc.
- Solutions: (see next page)



More Related Machine Learning Techniques

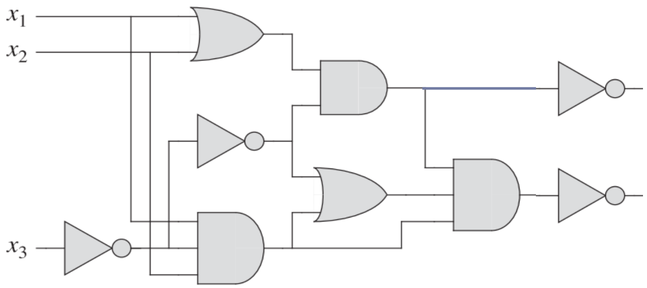
- Curriculum learning
- Transfer Learning
- Imitation Learning
- Behavior Cloning
- Dagger
- GAIL (like GAN)

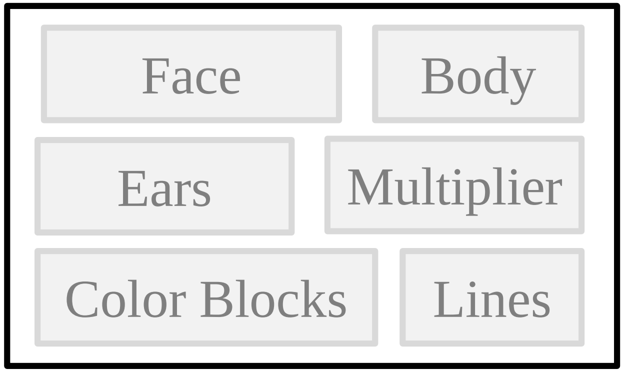
機會
Opportunities

挑戰 ▶

DL/RL/DRL in Practice

- 效果:
  1. 很多應用都證明帶來更高的品質
     - Like AlphaGo, Deep Q-learning, 2048 AI.
  2. 減少程式設計的複雜度與開發維護費用.
     - 很多if-then-else邏輯, 藏於DCNN中.





Face

Body

Ears

Multiplier

Color Blocks

Lines

  3. 很容易被應用於不同的問題
     - 例如: 非常容易將 2048 AI 轉成 Threes! AI. (無需改變heuristics)

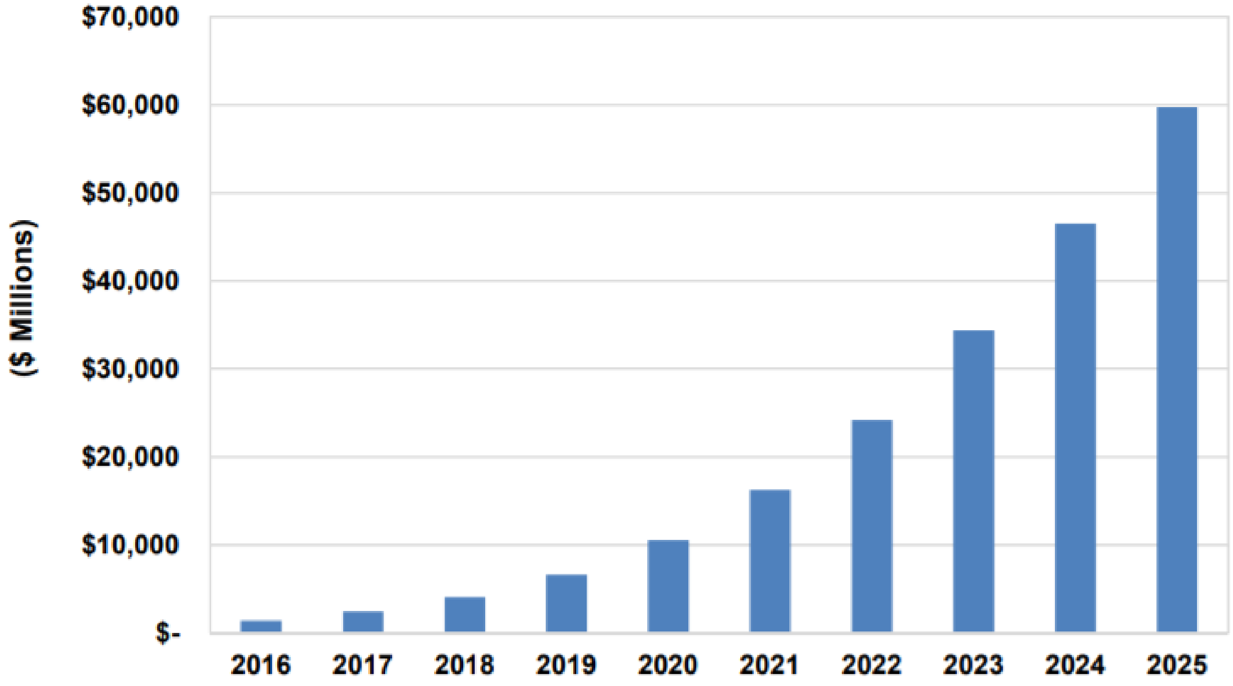
DL/RL/DRL技術的應用 – 產業

- 自動遊戲學習系統
  - 遊戲設計產業
  - 電競(eSports)產業
- 電腦視覺，語音聽覺，自然語言處理
  - 機器人、無人駕駛車、無人機
  - 故障偵測與分類(FDC)、Defect Detections
- Big Data
  - 消費、金融、教育、BI、防災
- 優化應用問題
  - 排程問題(例如:最佳路徑)、最佳涵蓋率(例如:廠房人員流量控管)、機組調度問題(例如:電廠省電)
- 醫療
  - 診斷、藥物分析、智慧居家照護
- 資訊安全
  - Hacker 攻擊

人工智慧市場大幅成長(Tractica,2017)

- 根據Tractica(2017)預估, AI軟體的直接間接應用之市場規模將從2016年的13.8億美元成長至2025年超過597.5億美元。

Chart 1.1 ***Artificial Intelligence Revenue, World Markets: 2016-2025***



(Source: Tractica)

Ongoing Research Topics at Our Lab

- Computer Games
  - Continue developing CGI
    - Combine “Zero” and multi-labelled value network.
  - Apply “Zero” to other computer games or applications.
    - Connect6 (or Gomoku), Chinese chess, Mahjong, ….
  - Work on Interpretability, leveraging “Zero”.
    - Combine heuristic and exact methods
- Industrial applications
  - AI bot for video games
  - Random bin picking in robotics grasping.
  - Surveillance
  - …

AI Bots for Video Games

- Techniques: DRL
- Applications:
  - Detect any weaknesses of a game after design.
  - Help run the games.

Surveillance –
Fusing Sensor and Visual Data (I)

- Monitor in-flow and out-flow trajectories

挑戰
Challenges

結語 ▶

大量運算支持

- DL/DRL需要大量的GPU作 training & testing.
  - DeepMind used 2000 TPUs for AlphaGo Zero
  - Our estimation:
    - At least 12000 GPU (GTX 1080 Ti)

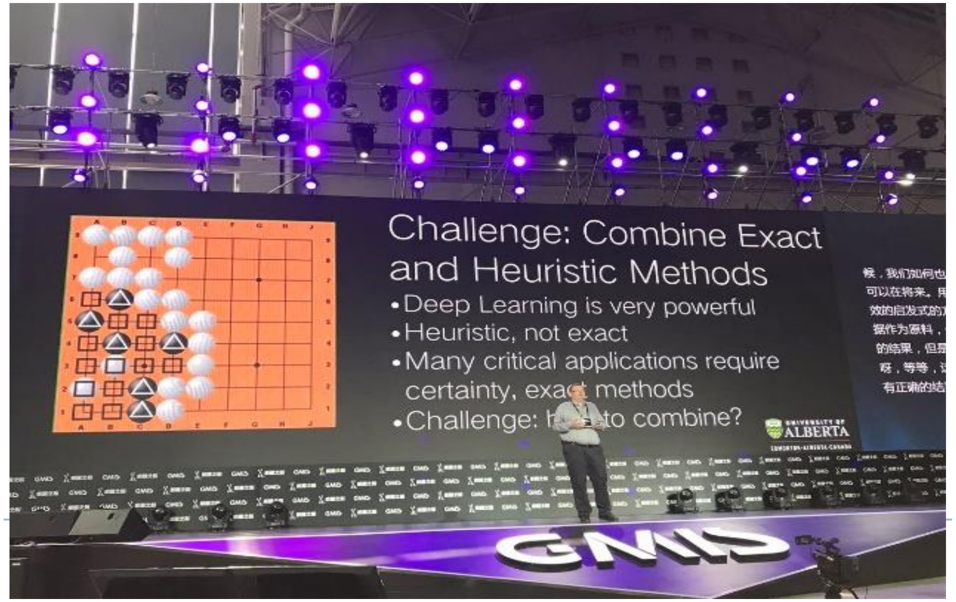
Challenges of DL/RL/DRL – in Training

- Need to tune carefully
  - Training data sets
    - e.g., the quality of data sets, generated by self-play?
  - Parameters:
    - e.g., learning rates, data/net sizes.
  - Weight initializations:
    - e.g., bias, Gaussian.
  - Nets in each layer:
    - e.g., filters, fully-connected, sub-sampling, ReLU (rectified linear unit), loss, max-pooling.
  - Solving overfitting:
    - e.g., L1/L2 regularization, dropout regularization, squeezing nets.
- Take a huge amount of time!
  Require huge amount of computing powers!

Need to try many cases.
These become **knowhow**!!

Interpretability (可解讀性)

- Importance of Interpretability
  - In many applications, we still need to interpret DNN.
    - Like medical problems.
- Martin Muller propose a new challenge:
  - “Combine Heuristic and Exact Methods?”

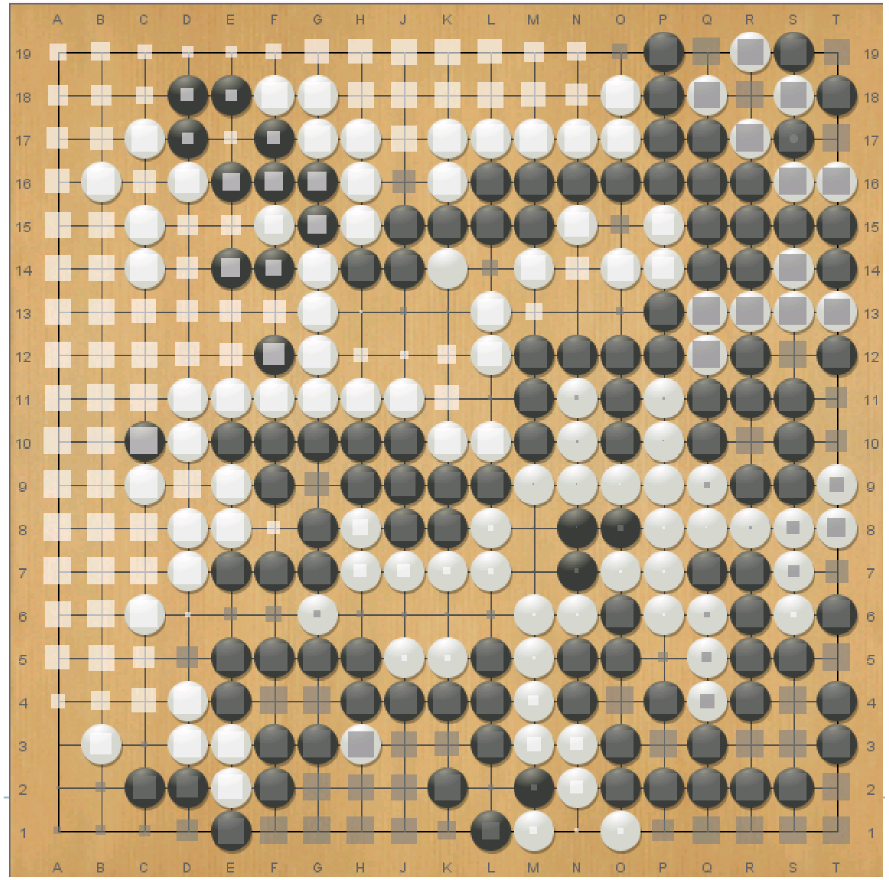


Challenge: Combine Exact and Heuristic Methods

- Deep Learning is very powerful
- Heuristic, not exact
- Many critical applications require certainty, exact methods
- Challenge: how to combine?

長龍問題: [Flyhigher]vs[絕藝] 2017/05/18

- 中下方白龍沒有兩眼
  - 但程式誤認白活, 以至於輸棋
    - DNN網路問題?
    - 其他問題?
    - 如何解決這問題?
- 若X光檢測出錯?
  - 任何安全做法
    可以協助保護?



結語

Conclusion and Acknowledgement

Conclusion

- AlphaGo (or DeepMind) demonstrates
  - The power of DL/RL/DRL. They are the future!!
- Challenges of DL/RL/DRL
  - The training knowhow is critical.
  - Interpretability is also critical.
  - Computing power support is also critical!!
- Our Lab:
  - Continue research on computer games
  - Explore other challenging applications based on our experiences on computer games.
    - Video Games
    - Robotics grasping problem
    - Surveillance
    - …

感謝科技部(MOST)的支持

- 深度學習專案計畫
  - 深度學習在輔助人類學習對局遊戲之應用
  - 提供關鍵計算資源

感謝海峰棋院捐助

財團法人培生文教基金會暨國立交通大學
捐贈與合作台灣圍棋實力躍升計畫簽約儀式
張翼
張懋中
林文伯
自由時報

特別感謝 **海峰棋院(林文伯先生)**
贊助交通大學電腦遊戲與智慧實驗室

感謝聯發科技捐助

特別感謝 **聯發科技**
贊助交通大學電腦遊戲與智慧實驗室

感謝實驗室團隊

- Leader: 吳廸融
- Other team members:
  - 陳冠文、吳宏君、賴東億

- Old members
  - 藍立呈、廖挺富
- New members
  - 劉安仁、謝孝忠

Thank You for Invitation and Listening!

Q & A