

程式人



用十分鐘瞭解

《電腦到底是怎麼下棋的》

（寫在 AlphaGo 首次擊敗李世石的隔天！）

陳鍾誠

2016 年 3 月 10 日

2016 年 3 月 9 日

- Google 的 AlphaGo 圍棋程式首戰九段超一流高手，韓國棋王《李世石》。
- 第一場快到尾聲時，李世石竟然棄子投降！
- 中視的解盤直播者都還沒搞清楚狀況時，比賽結束了！

這是繼 1997 年

- IBM 的西洋棋程式深藍 DeepBlue 打敗世界棋王 kasparov 之後，最受注目的一場人機大戰。
- 因為圍棋是對電腦難度最高的棋，一但電腦攻下圍棋後，所有主流棋類的棋王都將會是電腦了！

當我正在寫這一篇的時候

- 李世石和 AlphaGo 正在比第二場
- 現在網友們紛紛說李世石這場會輸
- 不過我覺得這並不準！

因為昨天第一場的時候

- 盤中一大堆網友都說李世石贏定了！
- 結果最後李世石棄子投降！

對於這種頂尖高手的對局

- 我想我們還是不要胡亂猜測的比較好！

特別是我

- 因為我根本看不懂，怎樣算贏怎樣算輸！
- 因為圍棋最後的輸贏還要看雙方各占多少地。
- 而我連最後占多少地怎麼算都不知道！

問題是

- 這樣我要怎麼告訴大家《電腦如何下棋》呢？

放心

- 山人自有辦法！

我雖然不太會下圍棋

- 但是會下五子棋和象棋
- 所以在這篇我會以五子棋為主，最後在稍微看一下圍棋！

其實

- 棋的規則並不那麼重要
- 電腦下棋所依靠的方法，和人腦大有不同！

不管是下

- 五子棋
- 象棋
- 西洋棋
- 還是圍棋

電腦所用的方法

- 其實都很像

只是每一種棋

- 對電腦而言，難度差很多

最難的是圍棋

- 最簡單的大概是五子棋了
- 而《西洋棋》和《象棋》的難度則是差不多，象棋可能稍難一點點！

這個難易度的排列

- 和《人對這四種棋的感覺》差不多！
- 但是、人和電腦兩者
 - 對難易的評判方法是完全不同的！

對人而言

- 一種棋很難，很可能是
 - 棋子種類很多
 - 或者規則很複雜
- 所以很難學！

但是對電腦而言

- 規則複雜其實沒有甚麼關係
- 電腦反正記憶力很強，而且很會遵守規則，規則太複雜只會讓人腦很難學，但是對電腦而言卻毫無影響，甚至反而會更有利，更容易贏！

所以、單就學習難度而言

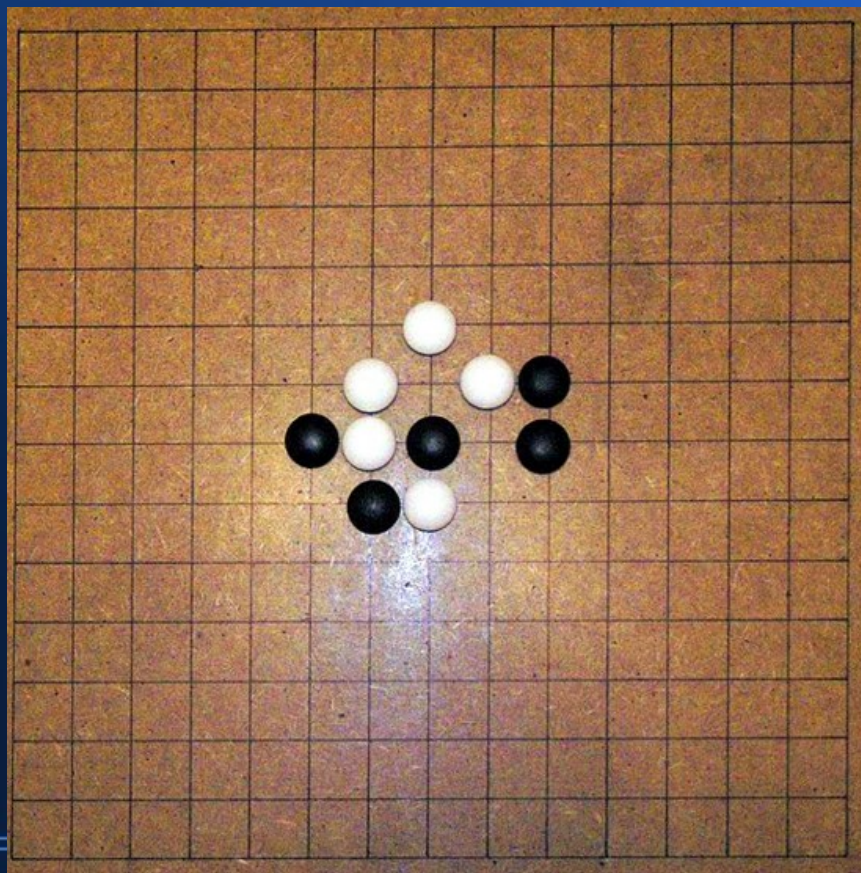
- 圍棋的規則有時還比象棋簡單
對人腦不見得更困難！
- 只是最後要算地計分，那部分我還沒搞懂！

好了、言歸正傳

- 那到底電腦是怎麼下棋的呢？

讓我們以最簡單的五子棋為例

- 來說明電腦下棋的方法！



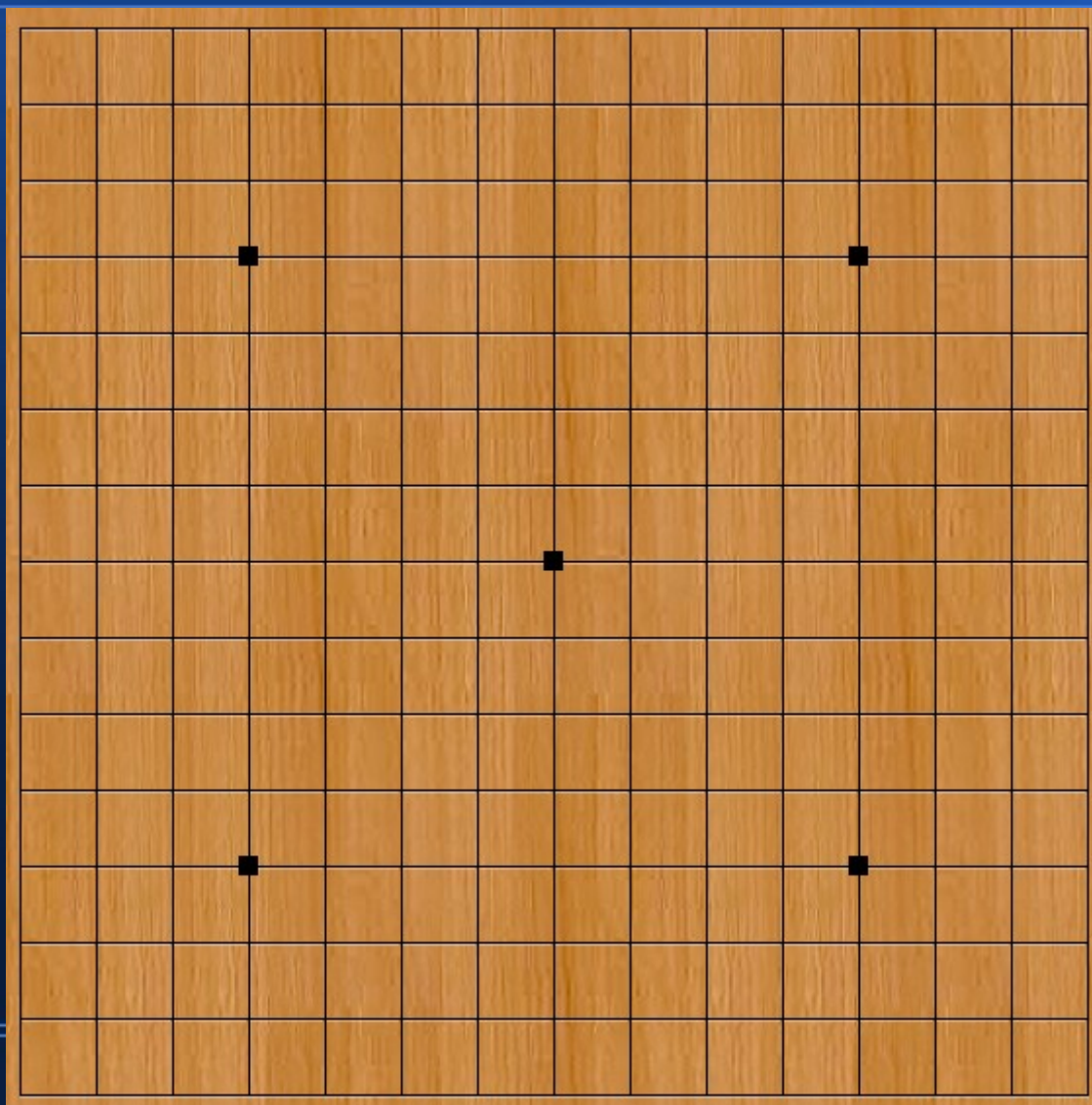
電腦下棋時

- 通常有兩個主要的關鍵算法
- 第一個是《盤面評估函數》
- 第二個是《搜尋很多層對局》，尋找最不容易被打敗的下法。

首先讓我們來看看盤面評估函數

- 讓我們用最簡單的五子棋為例，這樣比較好理解！

請大家先看看這個 15*15 的棋盤



注意：雖然格子只有 14×14 格，但五子棋是落在十字線上的，所以實際上是 15×15 個可以下的點。

如果不考慮最邊邊的話，那就會有 13×13 個可以下的位置。

不過以這個棋盤，邊邊是可以下的，所以應該是 15×15 的情況才對。

如果電腦先下

- 那第一子總共有 $15*15 = 225$ 種下法。
- 電腦下完後換人，此時還剩下 224 個位置可以下。
- 等到人下完換電腦，電腦又有 223 個位置可以下！

於是整盤棋的下法

- 最多有

$$-225*224*\dots*1 = 225!$$

- 種可能的下法

而且、這是 $15*15$ 的棋盤

- 標準圍棋棋盤是 $19*19=361$ 個格線，所以就會有 $361!$ 的可能下法！
- 只要能夠把所有可能性都確認，電腦就絕對不會下錯，基本上也就不會輸了！
- 但是 $361!$ 是個超天文數字，電腦就算再快，算到世界末日宇宙毀滅都還是算不完的！

不過、這件事情先讓我們暫時擱下

- 因為電腦就算算完了也沒有用，
重點是要算甚麼東西出來呢？

這個要算的東西

- 就是盤面評估函數！

以五子棋而言

- 我們可以用很簡單的方法，
計算目前盤面的分數。

以下是一個盤面評估函數的方案

- 連成 5 子：10000 分
- 連成 4 子：50 分
- 連成 3 子：20 分
- 連成 2 子：5 分
- 連成 1 子：1 分

等等、這只有考慮自己這方

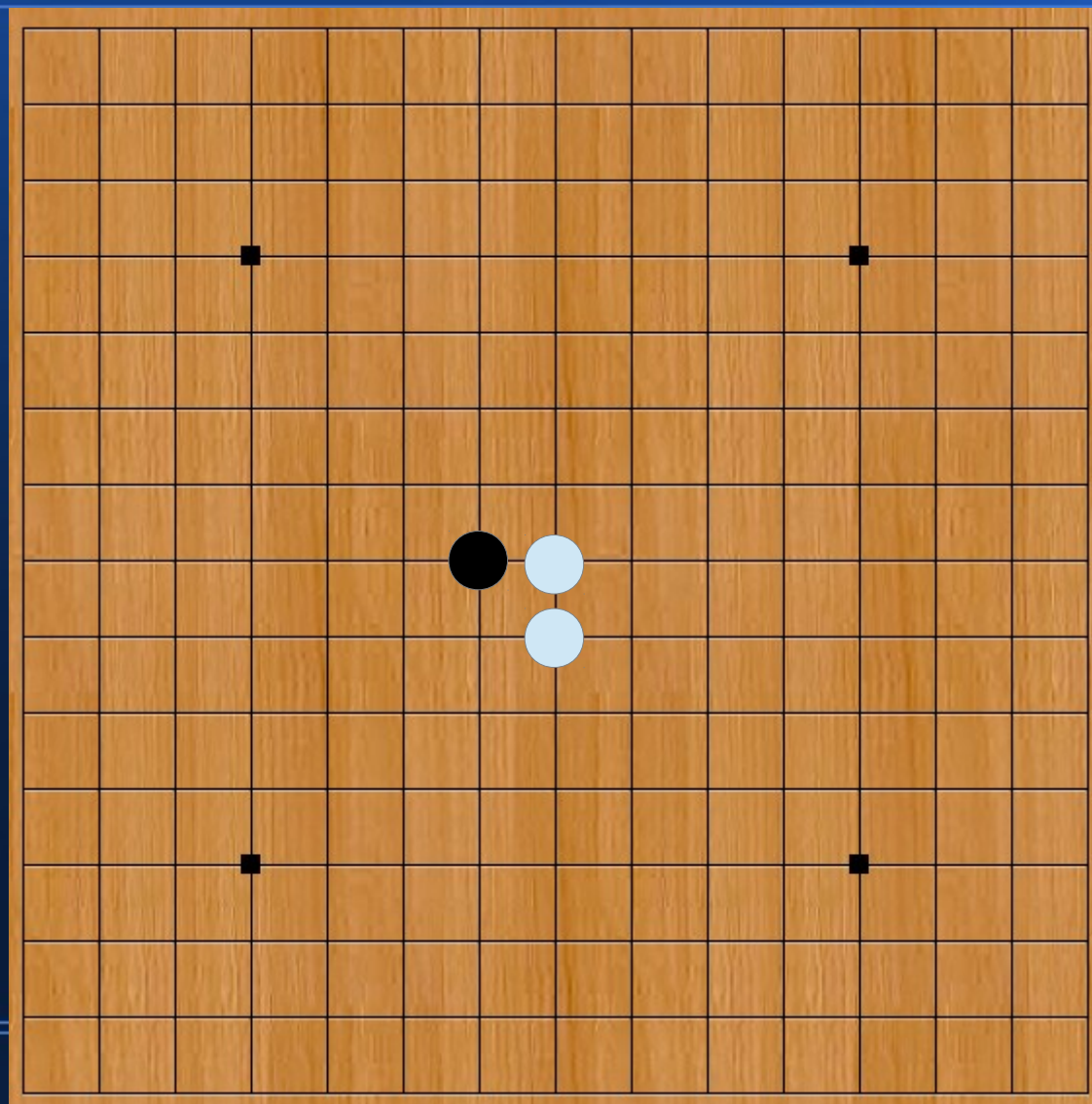
- 沒有考慮對方的得分！

沒錯

- 一個完整的盤面評估函數，應該考慮到雙方！
- 所以可以用
 - 我方得分 – 對方得分做為評估函數

舉例而言

連成 5 子：10000 分
連成 4 子：50 分
連成 3 子：20 分
連成 2 子：5 分
連成 1 子：1 分



在左邊的盤面中，假設電腦為白子。

白子兩顆連線，黑子只有一顆。

所以白子的得分為

$$5 + 1 * 2 = 7$$

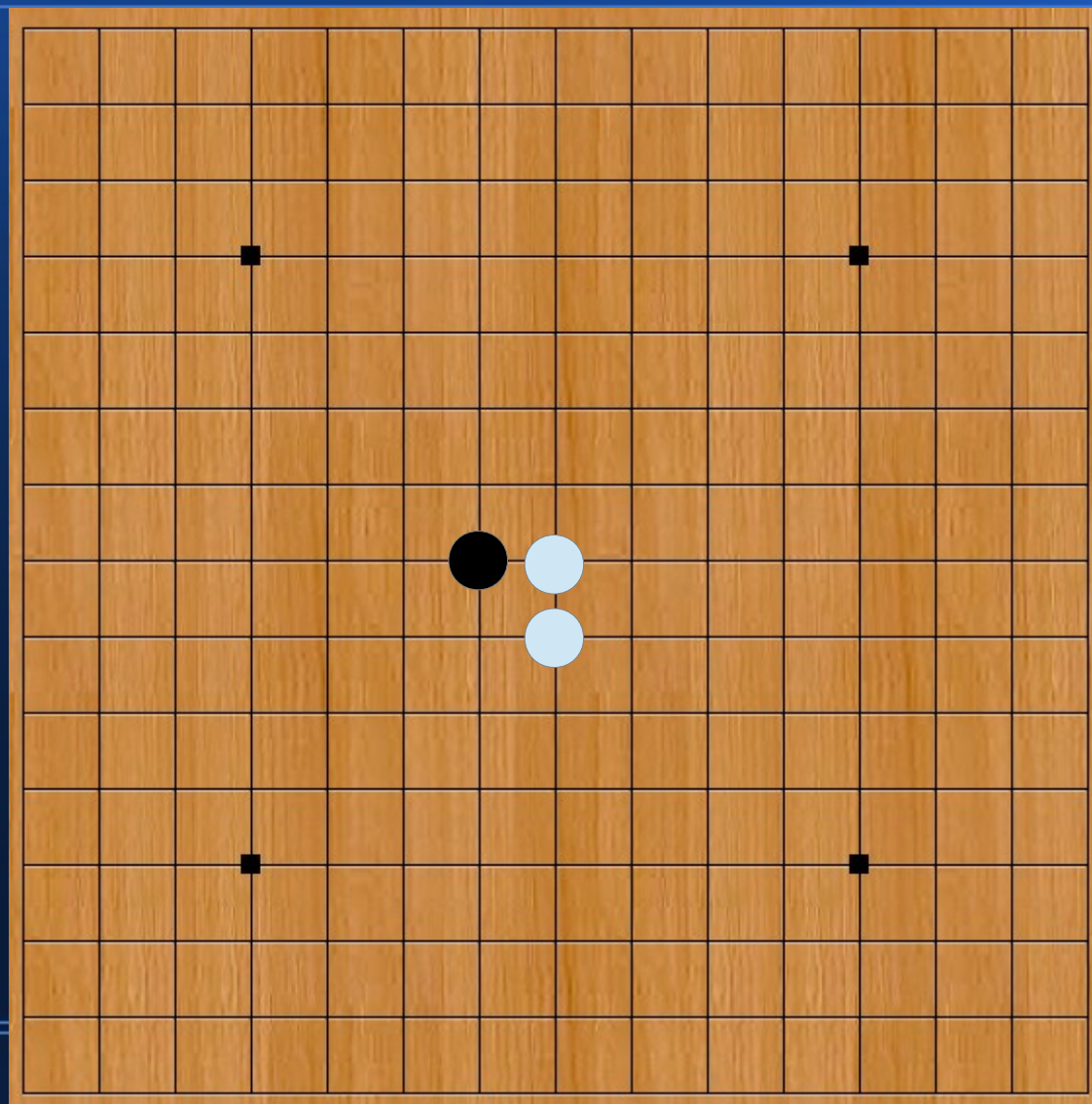
兩顆連線
得 5 分

兩個一顆的情況也計入
各得 1 分

雖然這裡有點重複算，但由於分數的設計差距夠大，所以沒有關係。

由於黑子只有一個 目前只得一分

連成 5 子：10000 分
連成 4 子：50 分
連成 3 子：20 分
連成 2 子：5 分
連成 1 子：1 分



所以對電腦而言，
盤面分數為

$$7-1 = 6$$

於是我們可以寫一個程式，
計算盤面的分數。

這個程式並不算難，對一個
學過基礎程式設計，會用
二維陣列的人應該是很容易
的。

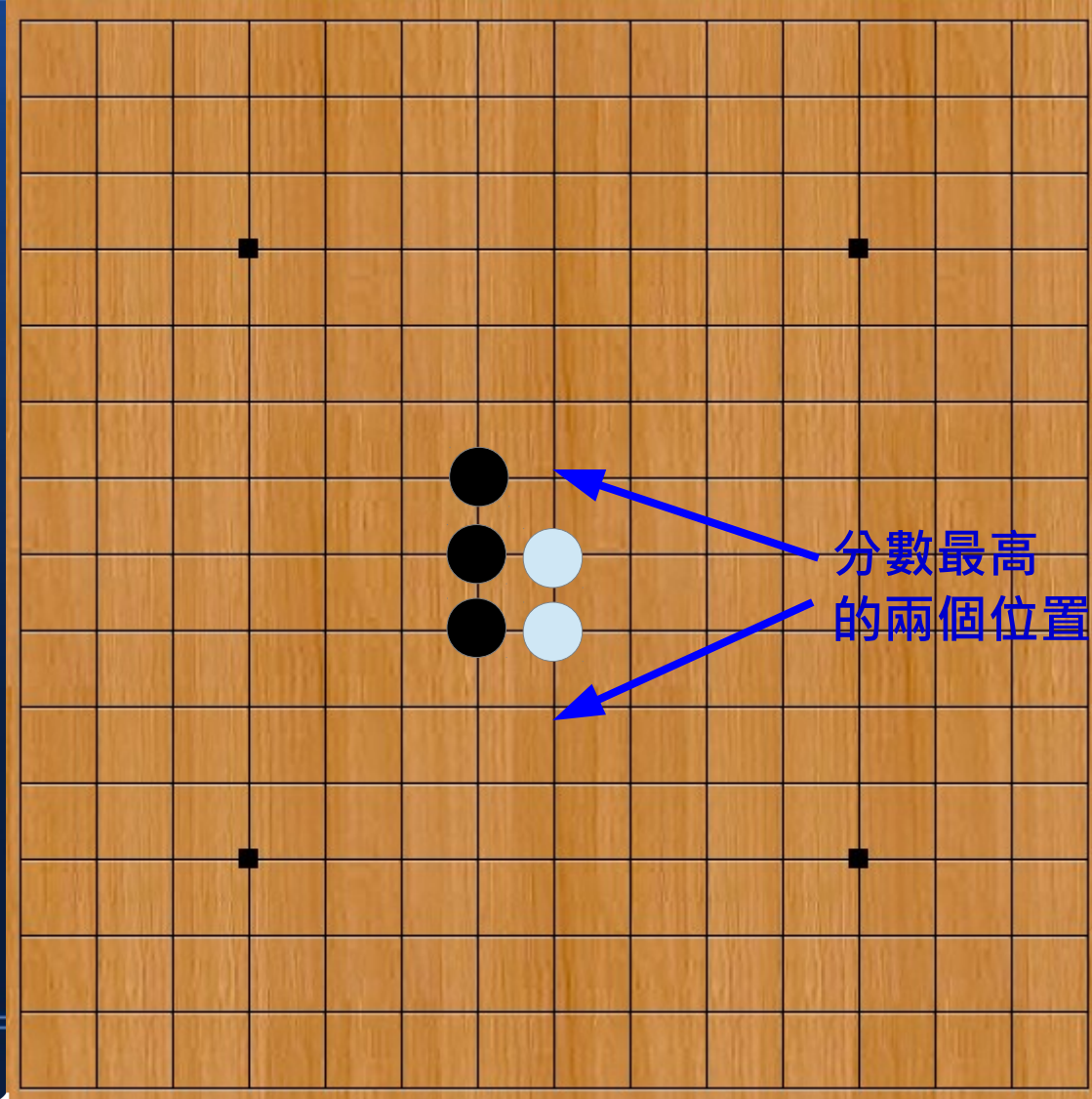
假設這個程式為 `score(B)`，
其中的 `B` 代表盤面陣列。

有了這個盤面評估函數 `score(B)`

- 我們其實就可以輕易建構出一個簡單的下棋程式了。
- 因為電腦只要把每個可以下的位置，下子之後的分數算出來，然後下在分數最高的那一格，就可以了！

最簡易的下棋程式

1,1 1,15



舉例而言，假如電腦
為白子，現在換電腦下：

那麼電腦會笨笨的計算

(1,1),(1,2), ..., (1,15),
(2,1), (2,2), ..., (2,15),
...
(15,1), (15,2)...(15,15)

當中還沒被下過的位置，
每一格下完後的分數。

然後挑出最高分的位置下子！

15,1

15,15

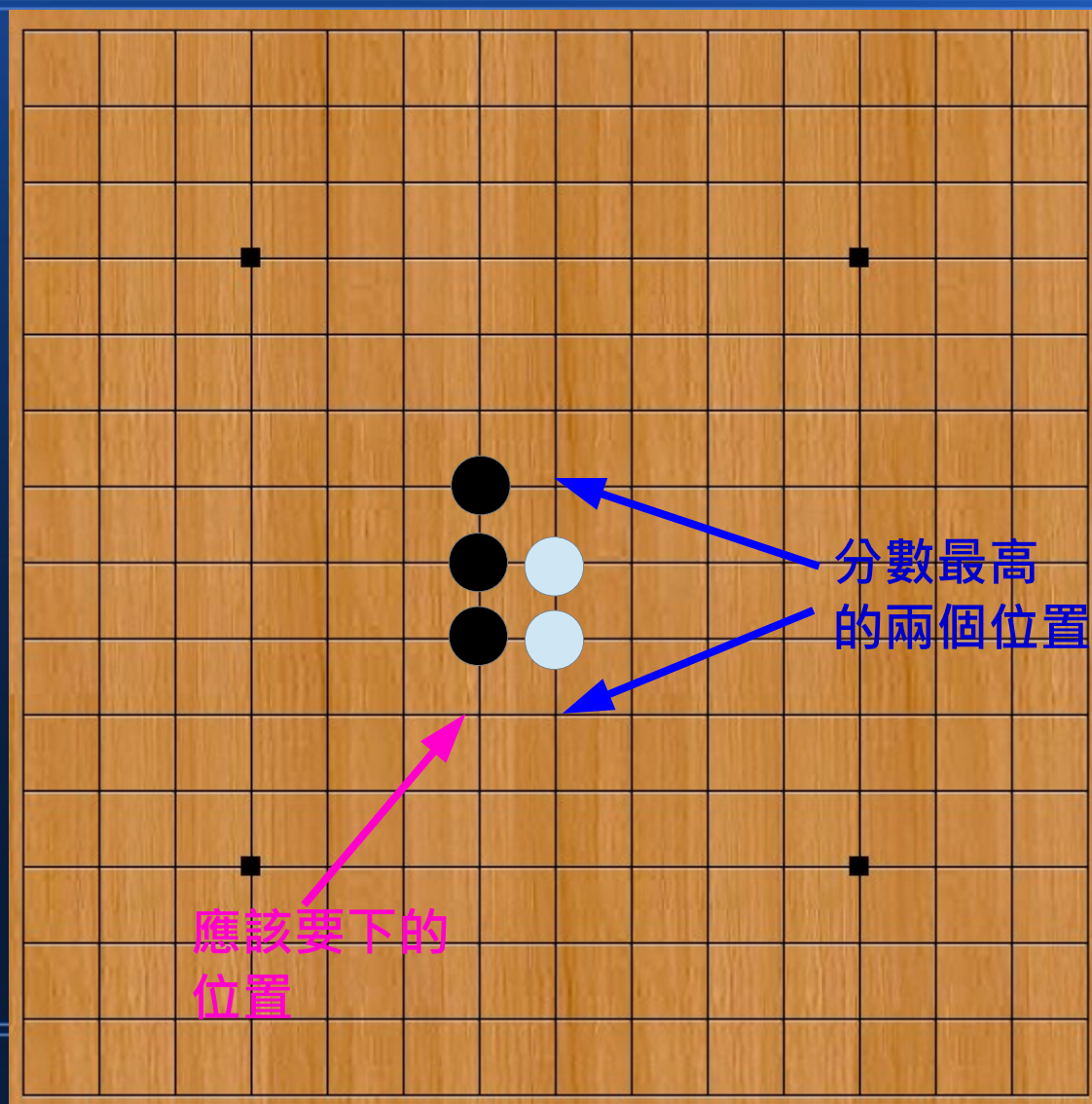
但是

- 這種程式的棋力不強
- 不過已經有可能下贏小孩或
棋力很弱的人

只是由於太過貪心

- 該程式只看自己的分數，不看對方下一手的分數。
- 如果你稍微做個洞給他跳，很容易就會贏了！

舉例而言



在左邊這個局面，如果輪到代表電腦的白子，那麼電腦會選擇下藍色箭頭所指的那兩個位置之一。

但是對五子棋有概念的人都會知道，這時候應該要下粉紅色箭頭所指的位置，否則就會輸了。

為了避免這個問題

- 電腦除了考慮攻擊的得分之外
- 還應該考慮防守的得分。

但是即使如此

- 棋力也不會太強，大概只能
下贏初學的小孩！

要提升電腦的棋力

- 就必須加上《對局搜尋》的功能！

到底

- 《對局搜尋》是甚麼呢？

更明確的說

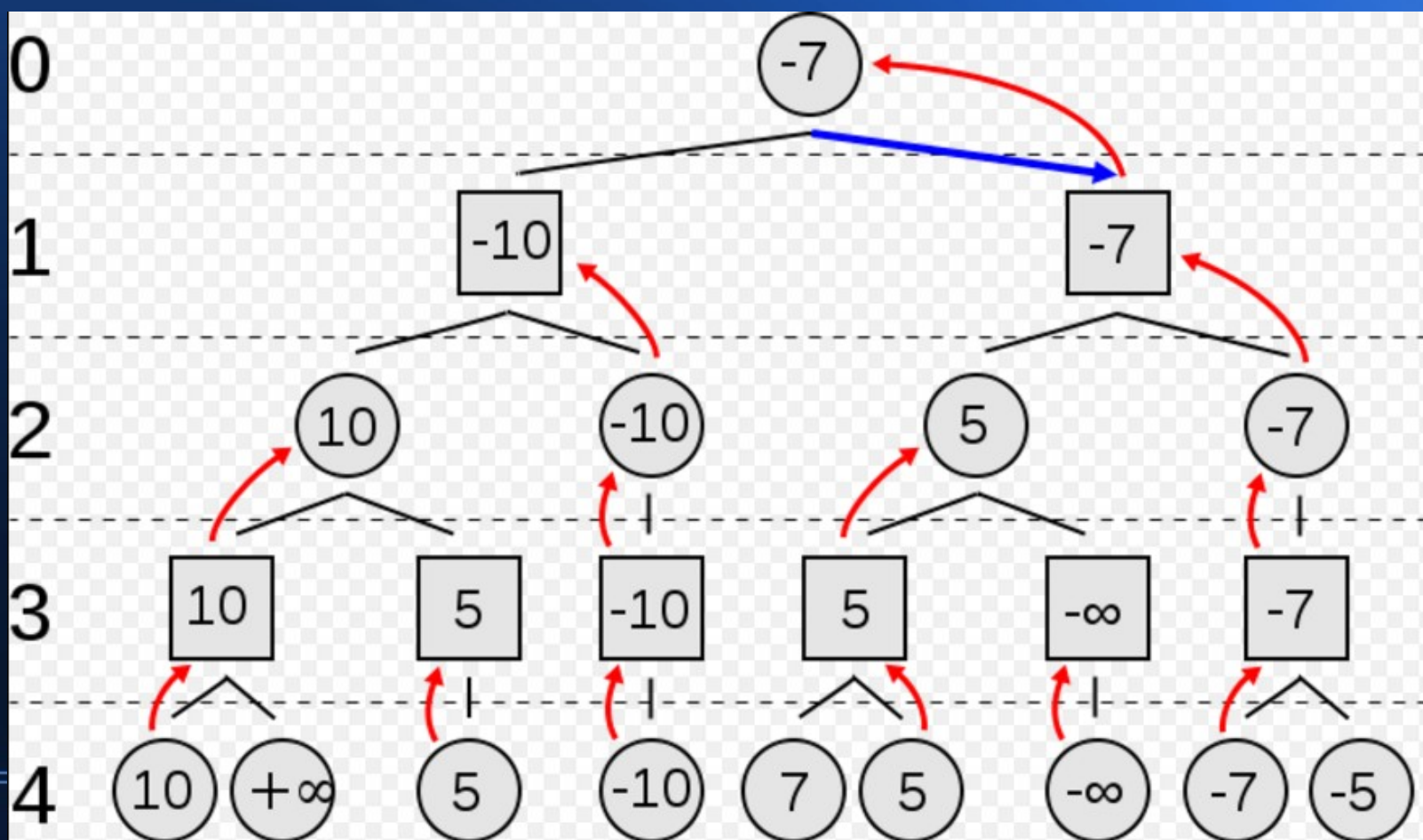
- 就是 MinMax 《極小極大》 演
算法

以下、讓我們圖解一下

- MinMax 演算法的想法！

下圖中的偶數層，代表我方下子 奇數層代表對方下子

我們必須找一個《最糟情況失分最少的路》，這樣在碰到高手時才不會一下被找到漏洞而打死！



但是、這樣的方式搜尋不了多少層！

- 因為如果每步有 $19 \times 19 = 361$ 種可能，那麼
 - 兩層就有 13 萬種可能
 - 三層就有四千七百萬種可能
 - 四層就有一百六十億種可能
 - 五層就有六兆種可能

電腦再快也無法搜尋超過十層

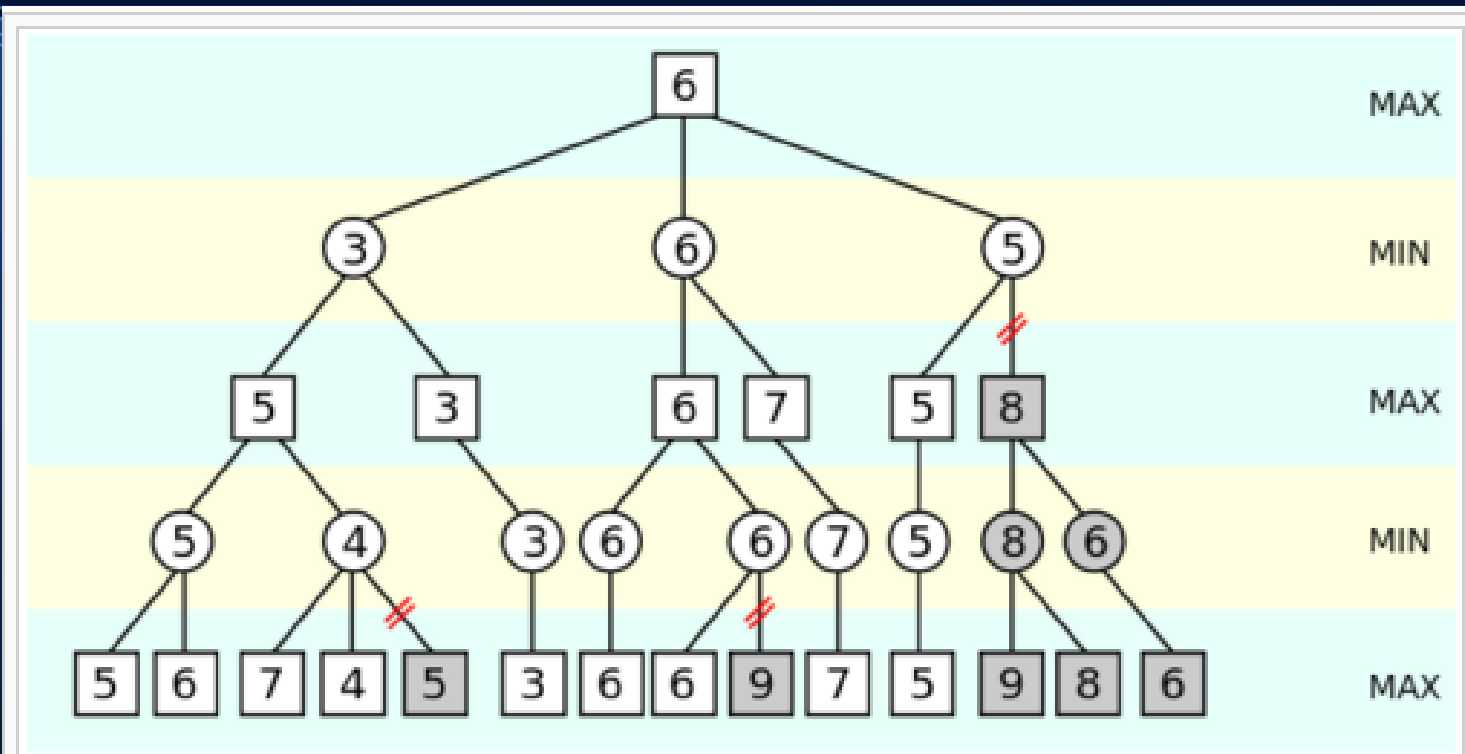
所以

- 還需要一些其他的方法，才能搜尋得更深！

這時候

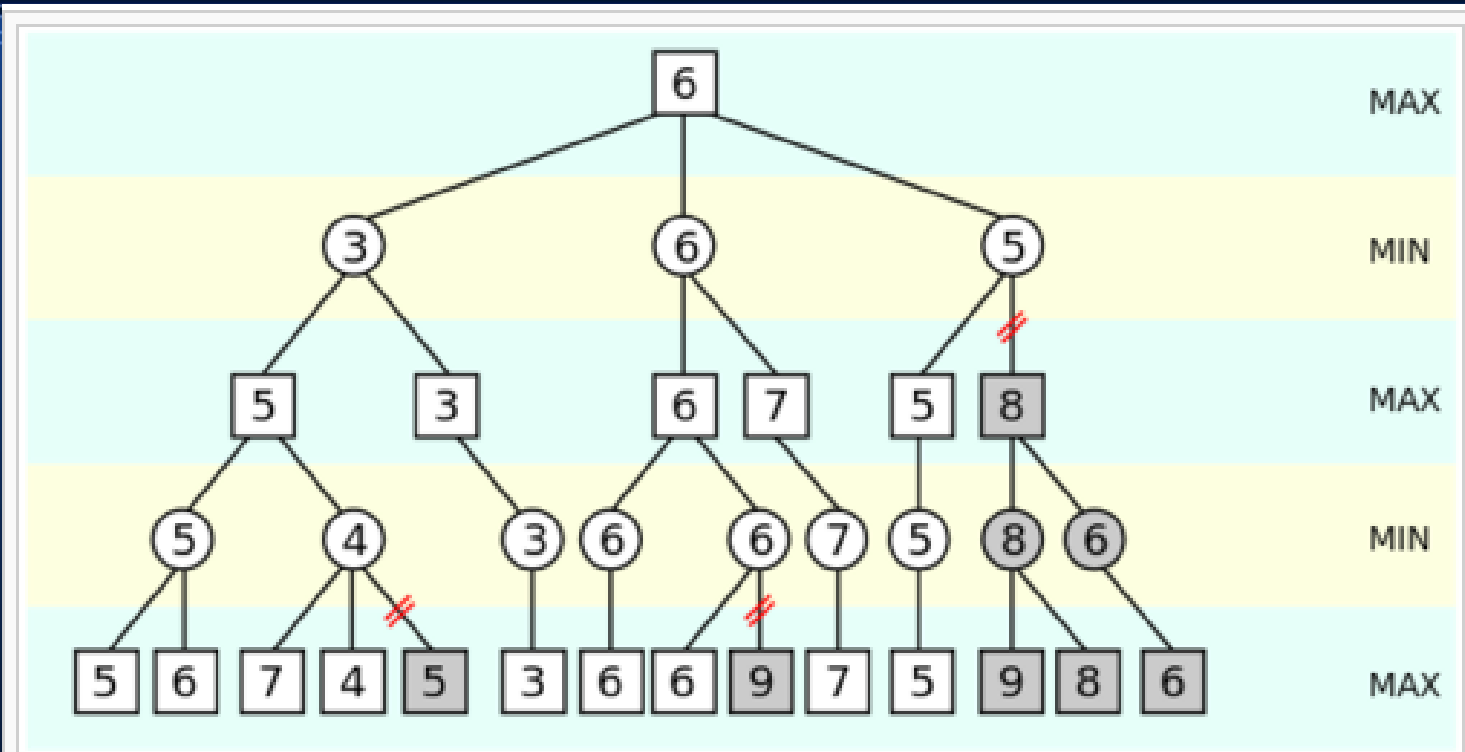
- 可以採用一種稱為 Alpha-Beta 修剪法的演算法
- 把一些已經確定不可能會改變結果的分枝修剪掉。
- 這樣就可以減少分枝數量，降低搜尋空間

以下是 Alpha-Beta 修剪法的範例



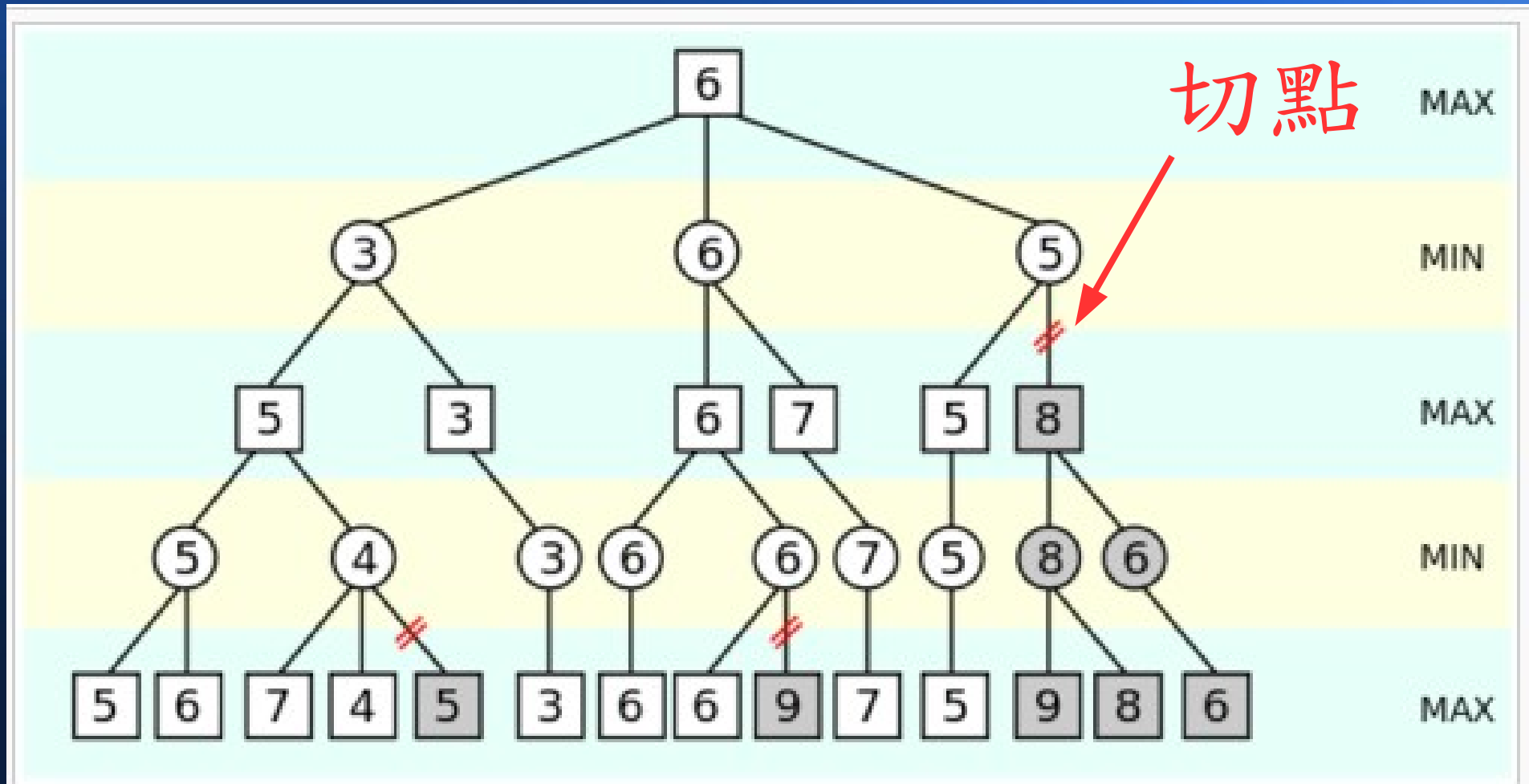
An illustration of alpha-beta pruning. The grayed-out subtrees need not be explored (when moves are evaluated from left to right), since we know the group of subtrees as a whole yields the value of an equivalent subtree or worse, and as such cannot influence the final result. The max and min levels represent the turn of the player and the adversary, respectively.

您可以看到雙紅線切掉的部分
就是 Alpha-Beta 修剪法的功效



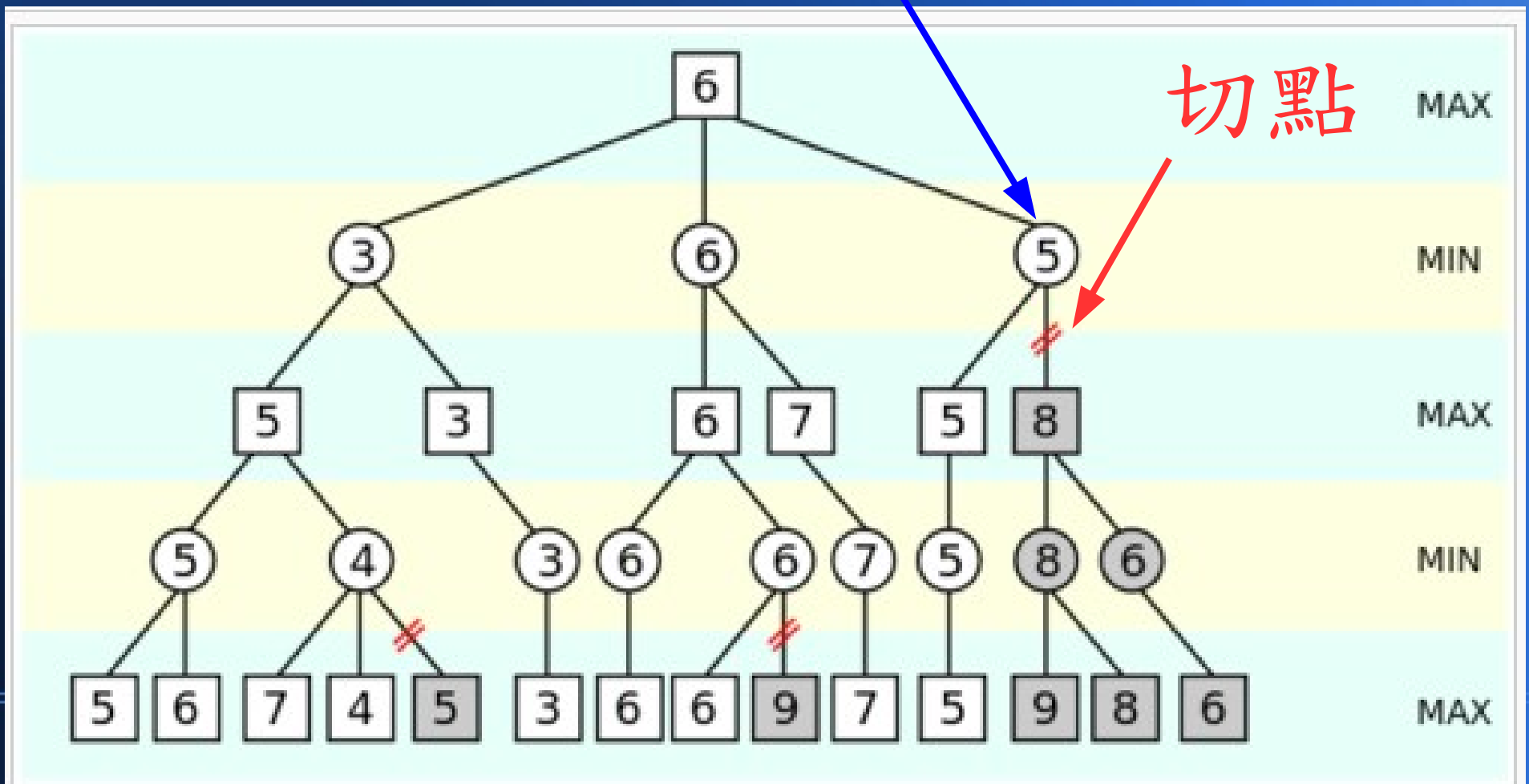
An illustration of alpha-beta pruning. The grayed-out subtrees need not be explored (when moves are evaluated from left to right), since we know the group of subtrees as a whole yields the value of an equivalent subtree or worse, and as such cannot influence the final result. The max and min levels represent the turn of the player and the adversary, respectively.

讓我們以圖中的切點為例
說明為何該部分可以切掉



因為切點上面 Min 層目前值為 5 比前面的 6 還小

因此後面的值不管多大，都只可能讓此處的數值變得更小，不可能更大了
所以後面的所有分枝都將不需要再算下去，可以修剪掉了！



於是透過 Alpha-Beta 修剪法
就可以大大減少分枝數量

- 讓電腦可以在固定的時間限制內，搜尋得更深更遠。
- 於是棋力就可以提高了！

這個 Alpha-Beta 修剪法

- 是由 LISP 的發明人 John McCarthy 所提出，後來由 Allen Newell and Herbert A. Simon 兩人實際用在下棋上。
- 這三位後來都曾經得過圖靈獎！

有了 MinMax 的搜尋

- 加上 Alpha-Beta 修剪法，電腦在五子棋上就可以輕易地擊敗人類了！
- 在西洋棋和象棋上，則還需要棋譜來訓練出更強更好的評估函數！

現在 Google 的 AlphaGo 程式

- 我原本猜測應該有採用 Min-Max 搜尋和 Alpha-Beta 修剪。但是在評估函數上還加入了《神經網路》，而這個神經網路的訓練，我得到的訊息應該是採用《蒙地卡羅法》來處理的。
- 不過後來聽說《蒙地卡羅法》不是用來訓練神經網路做評估函數的，似乎反而是用來《取代對局搜尋》的。

以下是我和網友的對談 或許可以提供更多一點的訊息



潘為竣 人工智慧在圍棋上，之所以比象棋、西洋棋難度差距極大的原因，並不是棋盤格子多，而是因為圍棋贏的條件在比賽過程不明朗，而象棋、西洋棋只要把將帥或國王吃掉就好，所以本篇介紹的修剪法在圍棋上幾乎無效，AlphaGo真正的意義在於「深度學習」，只讓電腦知道遊戲規則和輸贏條件，一開始或發現有大失誤時餵電腦棋譜，剩下的就是電腦和自己下棋累積經驗，所以AlphaGo最終的評估函數並非人類給他的，而是靠電腦自身的經驗不斷調整而學得的。

讚 · 回覆 · 4 · 33分鐘



陳鍾誠 請問我可以把你這段話放入投影片嗎？

這樣對讀者會很有幫助！

讚 · 回覆 · 2 · 26分鐘



潘為竣 哇！作者親自回覆，好榮幸喔，當然可以

讚 · 回覆 · 1 · 22分鐘



陳鍾誠 潘為竣 太感謝了！立刻來加！

讚 · 回覆 · 3分鐘

另一位網友 Mark Chang

- 也提供了更進一步的資訊



Mark Chang 傳統蒙地卡羅搜尋範圍是借由模擬方式，要走到最後一步，搜尋範圍很大，傳統上都是讓圍棋專家寫死一些規則來減少收尋範圍，但這種方法有其限制，**AlphaGo**則是用類神經網路訓練，從大量高手棋譜中學習規則，再用增強式學習，自己跟自己對弈，不斷進化，把這些高手棋譜給融會貫通。在下棋的時候，用蒙地卡羅來推算下一步該怎麼走，並帶入類神經訓練後的結果，來減少收尋範圍。

接著我上網搜尋了一下 AlphaGo 演算法的資訊

- 找到了這些
 - <https://deepmind.com/alpha-go.html>
 - Training Deep Convolutional Neural Networks to Play Go
 - 這篇和 AlphaGo 方法可能類似
 - Mastering the Game of Go with Deep Neural Networks and Tree Search
 - 這篇是 google 的，可能是 Nature 那篇的前身！

很可惜的是

- Google 最近的那篇論文投到 Nature 期刊

Mastering the game of Go with deep neural networks and tree search

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel & Demis Hassabis

[Affiliations](#) | [Contributions](#) | [Corresponding authors](#)

Nature **529**, 484–489 (28 January 2016) | doi:10.1038/nature16961

Received 11 November 2015 | Accepted 05 January 2016 | Published online 27 January 2016

我們得要付錢才能看

READ THE FULL ARTICLE

Subscribe to
Nature for full
access:
£210

Subscribe

ReadCube
Access*:
\$4.99 rent
\$9.99 buy

*printing and sharing
restrictions apply

Buy/Rent now

Purchase article
full text and PDF:
£22

Buy now

Already a subscriber? [Log in](#) now or [Register](#) for online access.

Additional access options:

[Use a document delivery service](#) | [Login via Athens](#) | [Purchase a site license](#) | [Institutional access](#)

自從我愈來愈瞭解學術體系的運作規則

- 開始反對《教育部、論文、期刊、教授評鑑》之後

就下定了決心

- 不投論文給要錢的期刊
- 絕不付錢給期刊買論文
- 自己的文章就放網路，例如自己網站或 SlideShare 上就好了！

因為

- 那些期刊都是死要錢的邪惡組織
- 拿學者的論文來出版，還向學者和讀者兩邊收錢，最後還把持論文商業權，作者自己都不能放上網路。

對這件事情有興趣的朋友 可以看看下列文章

https://www.twreporter.org/a/elsevier-vs-sci-hub

view B 網游之修羅傳說 - VI... B 網游之風流騎士 - 第... 功夫派, 功夫派灵... 4399生死狙击 双线... 光軍戰記-屠魔

國際時事星期二

推倒貪婪期刊付費高牆！學術界揭竿而起

文 / 楊芬瑩 2016.3.8

這幾年，國際學術圈出現一群「知識界的羅賓漢」，他們要從貪婪的學術期刊出版集團，重新取回知識的火種。

2011 年哈薩克 22 歲研究生艾爾巴金 (Alexandra Elbakyan) 寫程式設法取得昂貴期刊內容，供人免費下載，服務像她一樣窮酸的學生、科學家。這個名為 Sci-Hub 的網站開張後流量驚人，因而遭出版巨頭天價索賠，美國法院勒令撤銷網址，但艾爾巴金堅信，科學研究成果不應該私有化，Sci-Hub 很快重起爐灶，她的無畏無懼，讓媒體、支持者封她為「科學界的羅賓漢」。

該文網址如下

- <https://www.twreporter.org/a/elsevier-vs-sci-hub>

抱歉！

- 話題岔開太遠了！
- 讓我們言歸正傳，回到電腦下棋的主題上。

這次 AlphaGo 與李世石的對局

- AlphaGo 可以在首戰就獲勝，除了他們的《神經網路》的評估函數很強之外，《蒙地卡羅》隨機型算法應該也扮演很重要的角色！
- 但是否有用到傳統的 MinMax 搜尋法和 AlphaBeta 修剪法我還無法確定，我得再確認看看！

我想

- 羅馬應該不是一天造成的。
- AlphaGo 的人工智慧也是建立在前人 70 年的研究基礎上所建立的！
- 像是 MinMax, AlphaBeta, 神經網路，
蒙地卡羅法都是前人發展出來的。

關於以上的描述

- 就是我對電腦下棋算法的理解！

就在我快寫完這份投影片時

- AlphaGo 在第二盤又擊敗李世石了！
- 所以現在的戰績是 2 比 0
- 整個比賽採 5 戰 3 勝制！
- 所以李世石只要再輸一盤，電腦就確定戰勝人腦九段超一流的選手了！

雖然

- 目前世界積分第一的
是中國的柯潔
- 但據說他的實力和李世石，也只是
伯仲之間而已！

所以我認為

- 電腦幾乎已經確定在圍棋成為棋王了。
- 既然圍棋是主流棋類裡最難的，那麼電腦在棋類已然完勝人類！

就算李世石後三場可以逆轉

- 電腦成為棋靈王也將只是時間問題了！

還有

- 如果您仔細看棋賽，應該會注意到一個人！

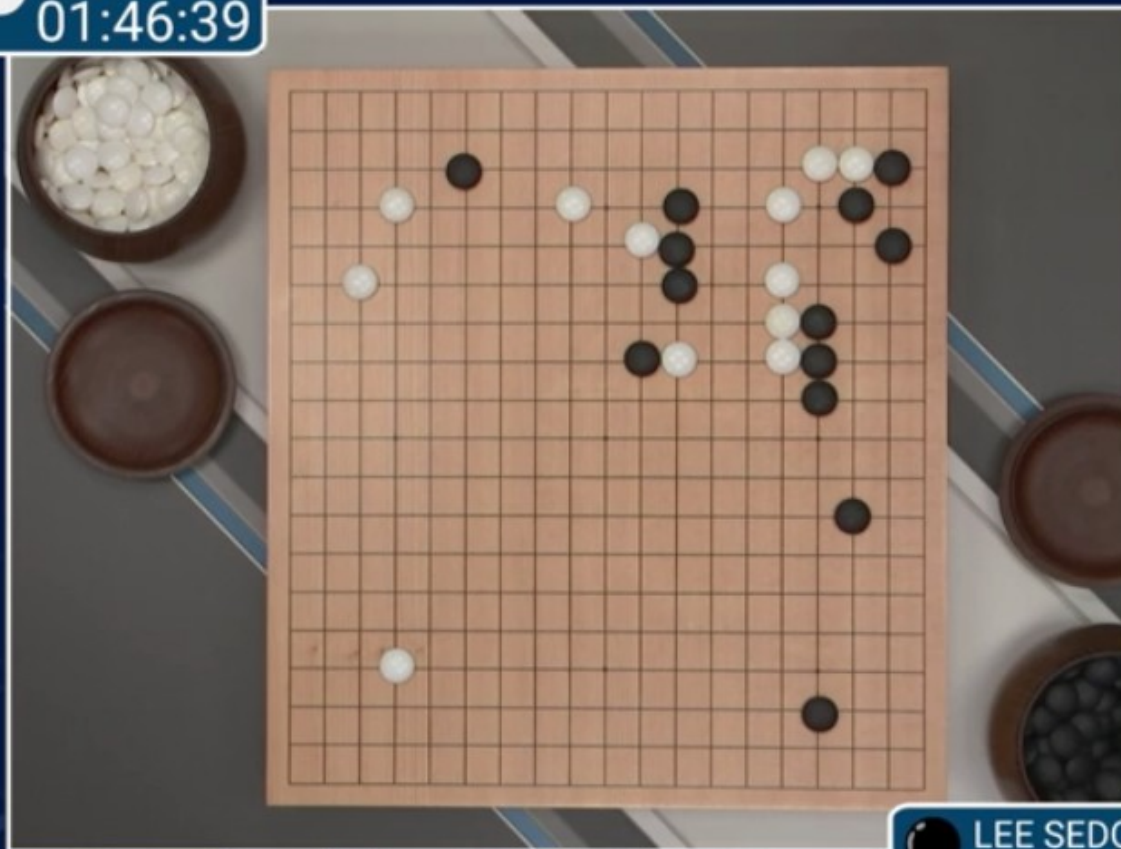
這個人在棋賽中就像是一台機器

而那台機器反而變得很像人

那個人就是

Match 1 - Google DeepMind Challenge Match: Lee Sedol vs AlphaGo ⌚ ➡ ⓘ

● ALPHAGO
01:46:39



● LEE SEDOL
01:47:17

 Google DeepMind
Challenge Match



他是誰

他是黃士傑

為甚麼他在那裏

Match 1 - Google DeepMind Challenge Match: Lee Sedol vs AlphaGo

ALPHAGO
01:46:39

Google DeepMind
Challenge Match



LEE SEDOL
01:47:17

那是因為

- AlphaGo 沒有手，沒辦法拿棋子！
- 所以黃士傑在那裏當他的手臂。

於是

- 機器變成了腦袋
人反而變成了四肢
- 機器指揮人來下棋！

但是、如果你以為黃士傑只是一隻手臂



台灣「土博」黃士傑（左）操作親手研發的AlphaGo，與南韓世界圍棋棋王李世石（右）對弈。
（美聯社）

那可就看走眼了

因為、黃士傑

- 可是個圍棋業餘六段的高手

而且也是研發 AlphaGo 的
重要推手

我們只要 看看他的博士論文就知道了！

論文基本資料	摘要	外文摘要	論文目次	參考文獻	電子全文
研究生中文姓名:	黃士傑				
研究生英文姓名:	Shih-Chieh Huang				
中文論文名稱:	應用於電腦圍棋之蒙地卡羅樹搜尋法的新啟發式演算法				
英文論文名稱:	New Heuristics for Monte Carlo Tree Search Applied to the Game of Go				
指導教授姓名:	指導教授：林順喜 指導教授：Rémi Coulom				
學位類別:	博士				
校院名稱:	國立臺灣師範大學				
系所名稱:	資訊工程學系				
畢業學年度:	99				
出版年:	2011				
語文別:	英文				
論文頁數:	91				
中文關鍵詞:	人工智慧 、 圍棋 、 電腦圍棋 、 蒙地卡羅樹搜尋 、 樹狀結構信賴上界法 、 模擬平衡化 、 時間控制 、 Erica				

黃士傑應該知道

- 他在圍棋上這輩子沒有機會
打敗李世石

所以、他決定研發機器大腦

- 用人機合一的方式
來打敗李世石

這兩天的棋局

- 證明了《人機合一》方法是
可以增強人類能力的！

但是、在這個同時

- 也可能會增強了機器的能力！

未來

- 或許戰局會變成

機器決定倒向哪一邊

- 哪一邊就會獲勝！

那如果

- 機器決定倒向恐怖分子或獨裁者呢？

原本、我們以為這只是一場比賽

現在看來

- 這已經不只是一場比賽了！

前天、我們的問題是

電腦到底能不能贏上一盤？

昨天、我們的問題變成

電腦和人腦到底是誰會贏？

今天、我們的問題卻是

人腦到底能不能贏上一盤？

明天、我們的問題會不會變成

人類到底還能不能繼續存活下去呢？

雖然、我們現在還能勝過電腦

因為我們只要做這個動作就行了

李師傅
不要跟他拼棋
嘗試切他電路

靠北工程師

但是一百年後

我們還有機會能切掉它的電路嗎？

人工智慧的進展

- 在最近 Google, Amazon, Tesla, IBM 等公司積極投入下，已然具有極高的商業價值。

未來的世界會怎樣

- 實在是難以預料阿！

不管如何

- 希望人工智慧帶來的是幸福，
而不是痛苦才好！

希望

- 這次的十分鐘系列投影片

能夠對您有所幫助

我們下次見

Bye Bye !