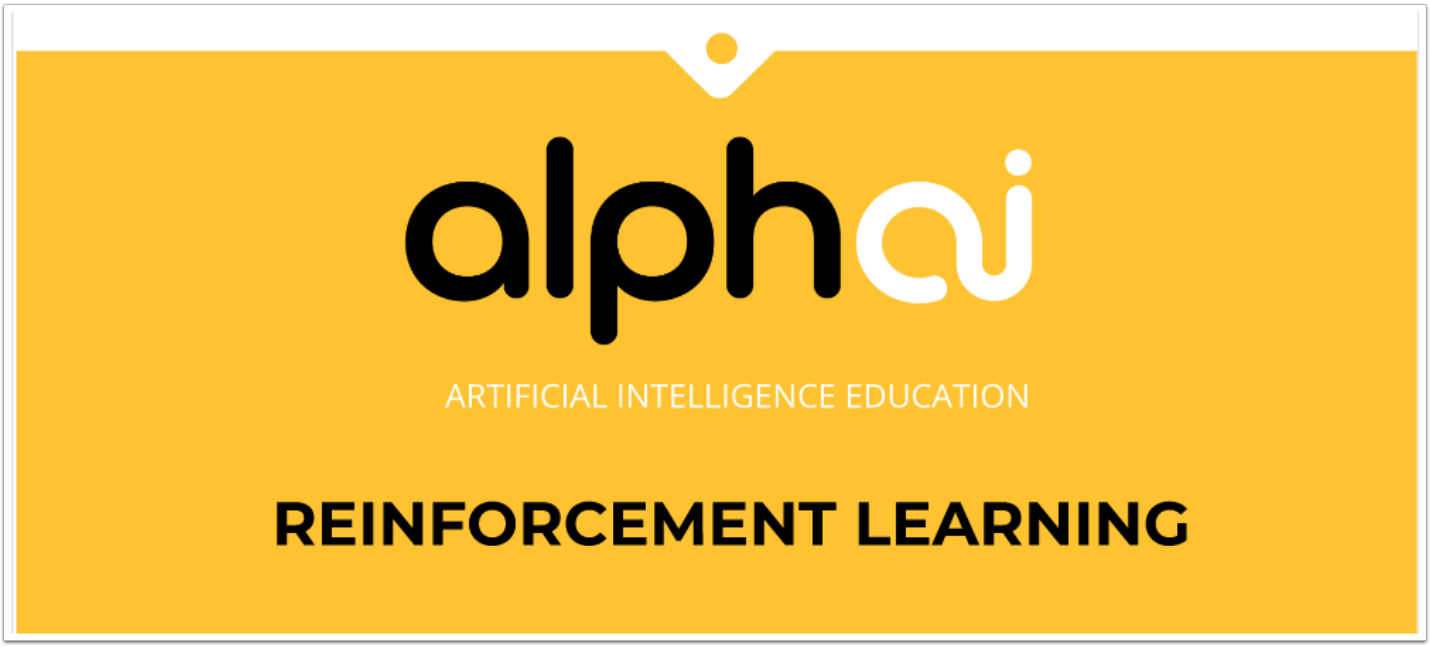
R2. 強化學習| Reinforcement Learning (advanced)



需要的知識：

- 神經網絡的基本概念

需要的軟件：

- 已經在電腦安裝 AlphAI software

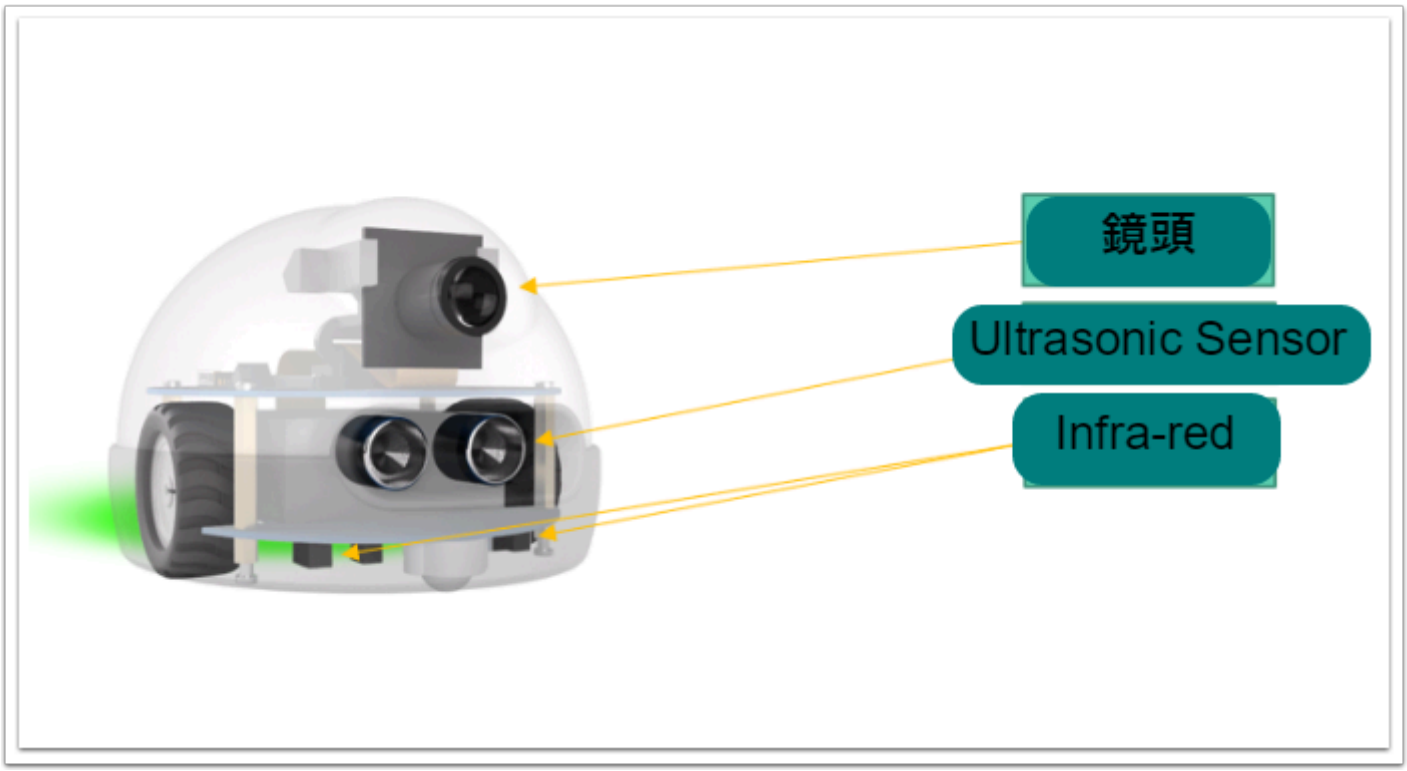
需要的硬件：

- 具備藍牙或Wi-Fi功能的電腦以和 AlphAI 進行連線
- 擁有 AlphAI 機器人及充電電池

ALPHAI 機器人 | THE ALPHAI ROBOT

AlphAI 機器人配備了很多 sensors (感應器)及 actuators(執行器)。它有：

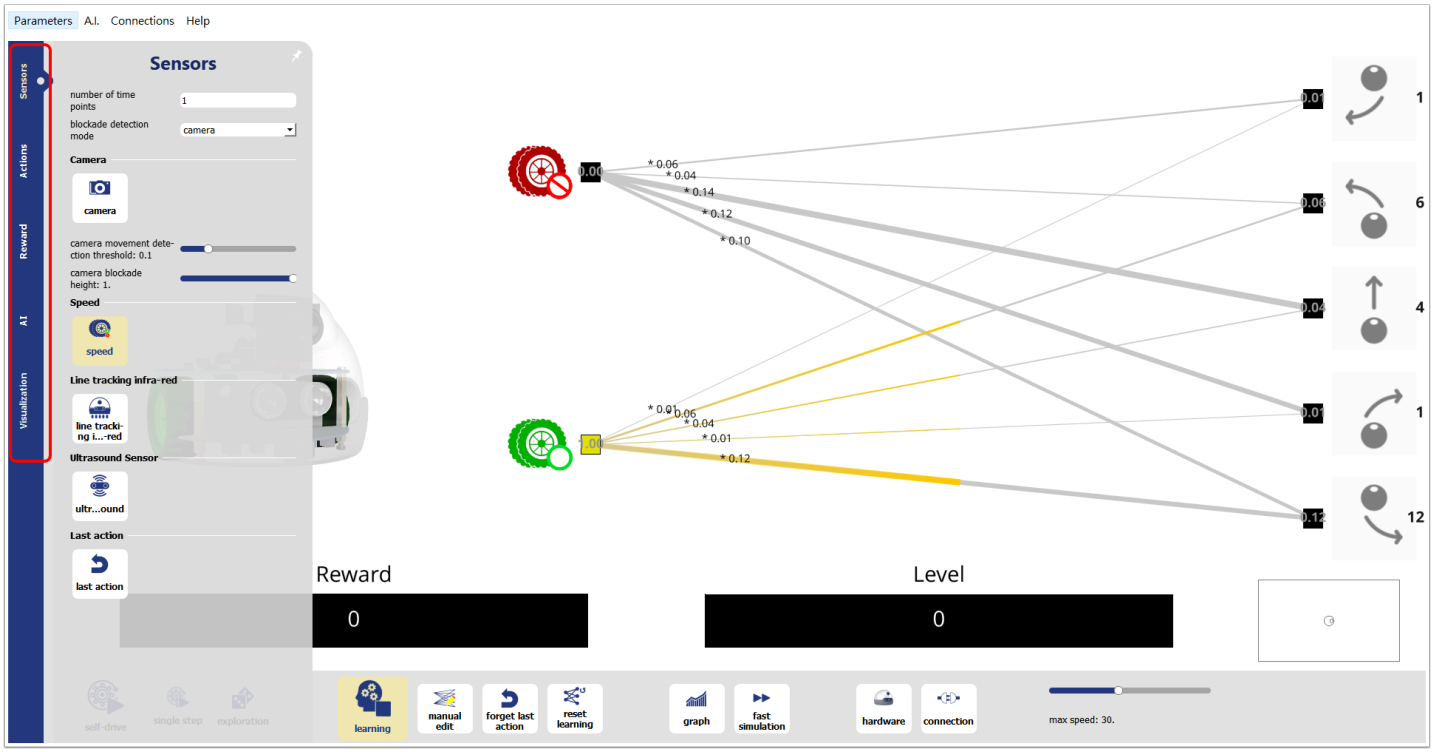
- 1 個 ultrasonic sensor(超音波感應器)測量正面距離
- 5 個朝下方的 infrared sensors(紅外線感應器) [在機器人底部]
- 1 個用於影像處理的攝錄鏡頭
- 2 個電動車輪(Motor)來進行移動
- 1 個 Buzzer (蜂鳴器)



它由微型電腦 Raspberry Pi Zero 控制，通過 Wi-Fi 或藍牙與 PC(電腦) 進行通信。AlphAI software **將向** AlphAI **機器人發出指令以進行決策，例如：「轉右」或「**Ultrasonic senosr **的狀態是什麼」等的指令。機器人將簡單地執行這些指令，真正的人工智能是在接下來的部份。**

ALPHAI SOFTWARE

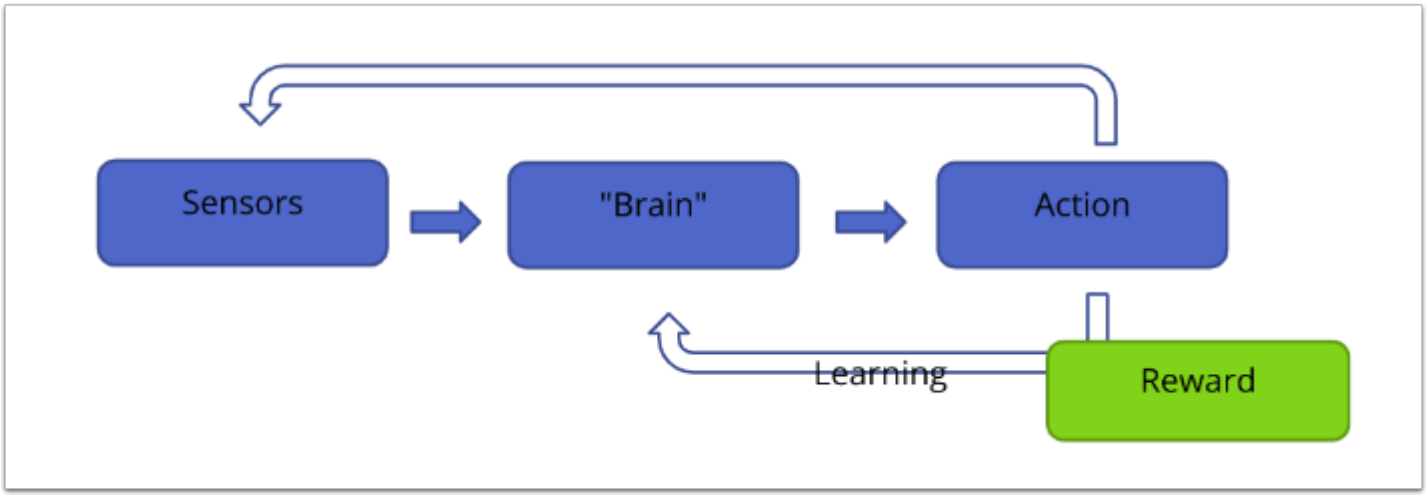
要控制 AlphAI **機器人，我們需用到「**AlphAI**」**software**。**



左面工具欄是設置 AlphAI 及各種學習的地方。

- **「Sensor」 讓用家選擇使用什麼 sensor 來進行學習**
- **「Action」則讓用家在經過演算法後選擇要執行那些不同的動作**
- **「Reward」讓用家調整機器人在學習過程中獲得的獎勵及懲罰**
- **「AI」對不同可用的演算法及參組進行分組並供用家選擇**
- **「Visualization」可調整圖像界面的不同視覺參數，例如：改變相機的解像度等等....**

引言：



本課節的目標是理解機器(機器人，或更一般的程式)如何「自行學習」：這稱為機器學習。

我們將特別提及強化學習及人工神經網絡。

強化學習是指讓機器人通過勵及懲罰系統從經驗中學習。(trail and error)

手動編輯網絡 | MANUAL NETWORK EDITING

引言 :

在第一部份中 , 我們將會看到一個迷你人工神經網絡如何透過 sensor data 做出行動決策。在當情況下是還未涉及機器學習 : 你需要在神經網絡中創建適當的連接以獲取最大的獎勵。

目的 :

- 創建正確的連接以最大化機器人的獎勵(reward)及等級(level)。

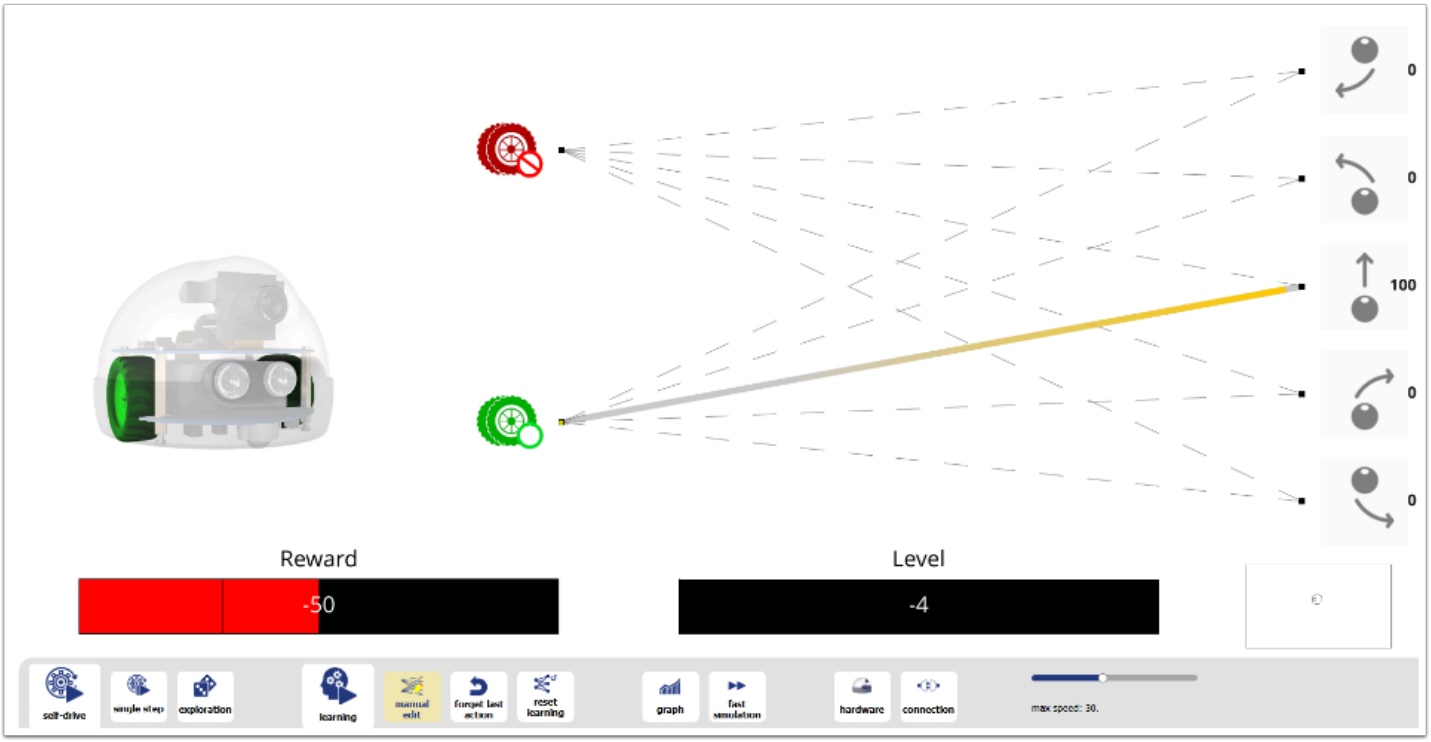
參數化 | Parameterization :

1. 開啟 AlphAI 的電源 (電源掣在底部)。需時約20-30秒 , 它會輕輕移動幾步, 當它準備好連接時會亮起白光。同時記下 AlphAI 底板的編號 (應該3位數 e.g 197)
2. 選擇以WiFi 或 藍牙接駁 (當超過4個 AlphAI 或房間已有多個 WiFi 網絡時 , 建議使用藍牙接駁)
3. 按「Connect」連接 AlphAI , 成功的話會看到電池電量。

Wi-Fi :	**藍牙 :**
- 將電腦連接至 AlphAI 的 Wi-Fi :找出以 AlPHAI 開頭並以機械人編號結尾的 Wi-Fi 網絡 : 密碼與Wi-Fi名稱相同 (包括大小寫)- 在「Tools」工具欄中選擇「wifi」	- 在 「Tools」工具欄中選擇「Bluetooth」,選擇相應的 AlphAI (對應機身編號)- 如果你的 AlphAI 不在列表中 , 點擊「pari a new robot via Bluetooth」並在該 AlphAI 出現時選擇 它 , 然後將它添加到您可以選擇的列表中。

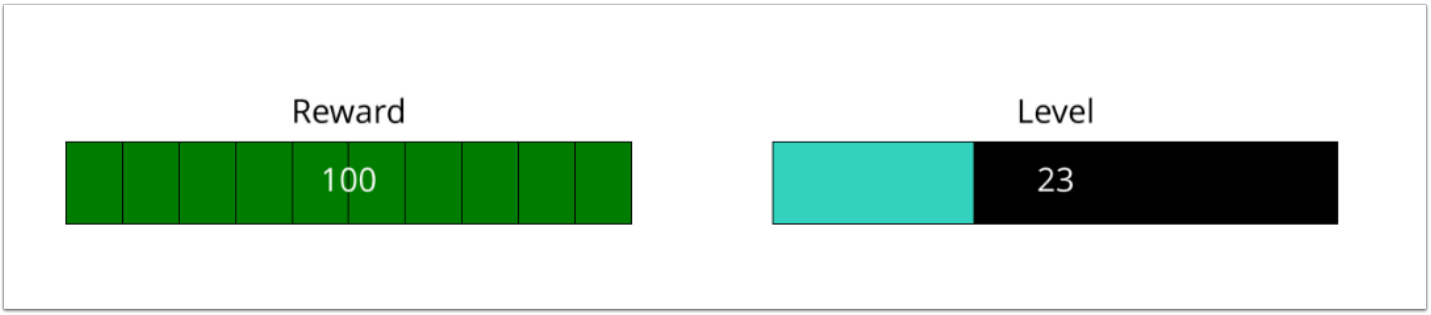
啟動 | Startup

1. 在「Parameters」清單中(dashboard左上角)點擊「load demo parameters」,然後選擇「Manual Edition Blocked vs. Motion」
2. 現在就開始吧！你將在機器人的神經網絡中創建連接 : 停止使用「learning」及「exploration」並開啟「manual edit」然後點擊「reset learning」。
3. 你在dashboard上應該會看到人工神經網絡的連接。這個網絡有7個神經元 : 2個輸入神經元(sensor)和5個輸出神經元(action)



4. 你可以透過點擊虛線以創建新的連接。這些連接會將輸入神經元的活動傳遞給輸出神經元。因此，它們將對應特定狀態(左面)而選擇特定的動作(右面)。在上面的示例中，我們創建的連接允許機器人在未被擋住的情況下選擇向前行駛這個動作。

5. 啟用「self-drive」mode : 綠色神經元(沒被卡住)亮起並會在你創建連接後立即激活一個動作。你可以創建正確的連接讓機器人獲得最大的獎勵(reward)。提示 : 機器人向前移動時獎勵為正數，向後移動或被障礙物卡住時為負數。



6. 觀察等級(level)的數值 : 它將顯示在過去一分鐘獎勵(reward)的平均值。注意它達到的數值：我們將與機器人自己學習所取得的等級作出比較。

機器學習 : 被阻 / 運行 | MACHINE LEARNING: BLOCKED / MOTION

引言 :

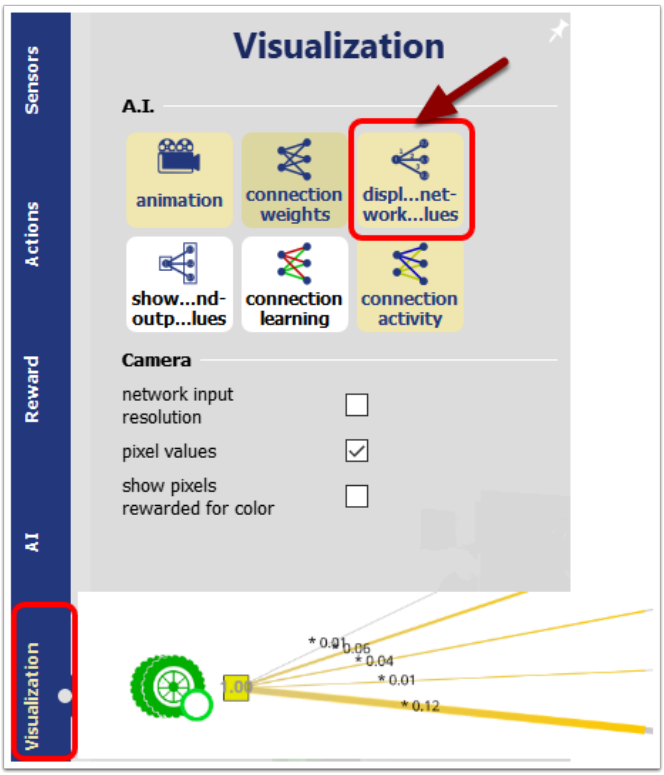
現在你已經了理機器人如何「選擇」動作(通過連接) , 讓我們看看機器人如何能夠學習選擇正確的動作並最大化獎勵。

目的 :

- 使用正確的設置讓機器人學習那些動作是最好的(最大化獎勵)。

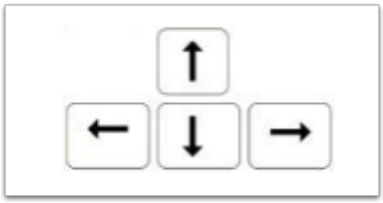
「無人駕駛機器人」 | "Piloted robot"

1. 關閉「manual edit」並開啟「learning」, 然後點擊「reset learning」 : 這會隨機初始化整個連接。



2. 在「Visualization」 tab 開啟「display network values」。你會在連線上看到一些數字。線的粗幼度代表連接的「強度」。試解釋為什麼顯示的連接並不「好」，並需在學習過程中「改進」？

3. 首先你需要駕駛機器人。它不會選擇自己的行動 , 只會學習。為此請禁用「self-drive」。



4. 機器人正在等待你來駕駛它！你可通透過點擊右面的箭頭或使用鍵盤的方向鍵來執行此操作。

5. 仔細觀察連接(線)的粗幼及行動上的數字 ,你覺得這些數字的作用是什麼？這有什麼含義 , 它是如何演變的? 它背後的計算是什麼？

說明你的想法吧！提示 : 這表示學習的結果

6. 當你認為學習/訓練已經完結 , 關上「learning」並打開「self-drive」。現在 , 機器人將自行移動 : 它的行動有否乎合預期？如果沒有 , 試重新開啟「learning」重新訓練機器人。

自主機器人 | Autonomous robot

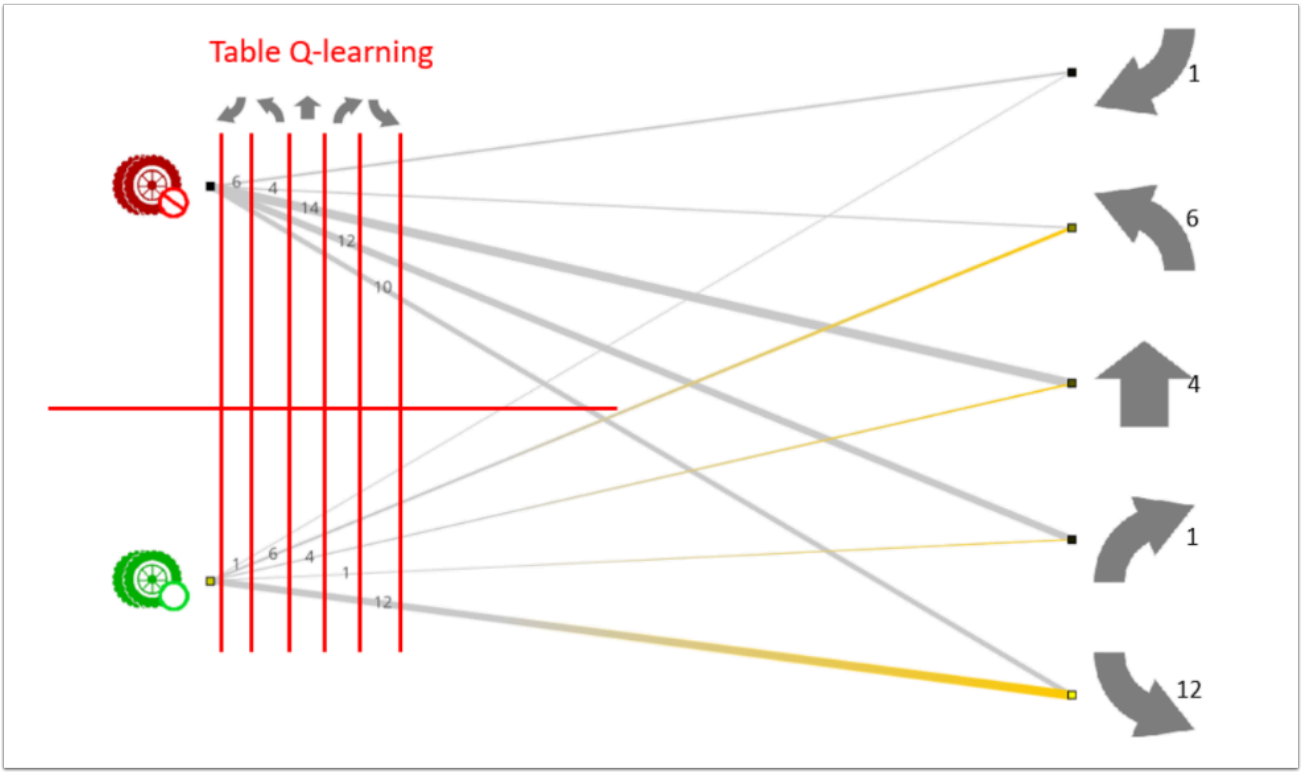
7. 再次開始訓練,這次讓機器人自行選擇動作(先按「reset learning」然後再按「self-drive」。你觀察到什麼？機器人有正確地學習嗎？又有什麼是缺少的？

8. 在底部工作欄開啟「exploration」並觀察學習是有什麼改變。這個參作有什麼作用？試說出你的想法。

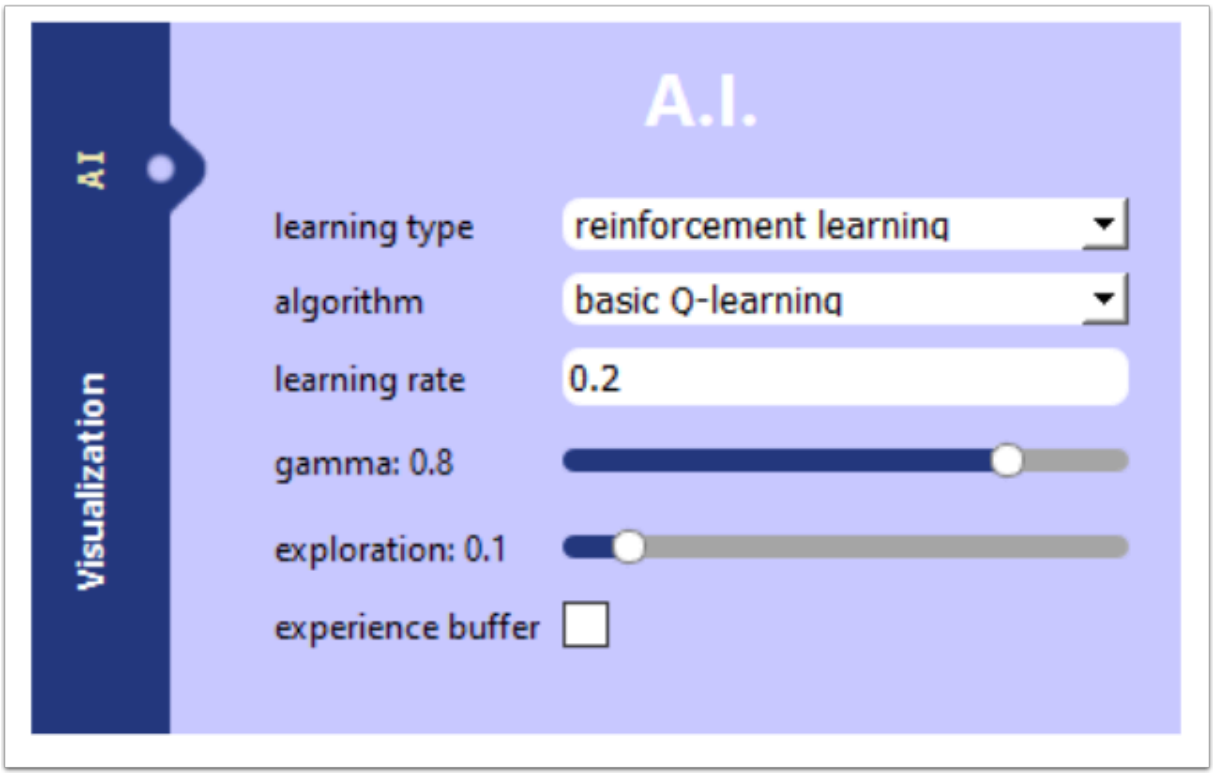
理解數學細節 | Understanding mathematical details

你應該已經觀察到訓練/學習會改變神經網絡的連接。繼續以下的步驟以了解關於公式(equation)的細節。你也可以直接進入第三部份開始配合相機進行學習，如果還有時間的話可以再返回此步驟。

這邊使用的演算法是 Q-learning，它是基於下圖所展示的數值。這個圖表通過數學計算來確定下一步將進行那個動作。列(columns)代表可能的動作，行(row)代表可能的狀態(一般利用sensors收集)。至於數值，它們對應狀態和動作之間的連接權重(weights of connections)。



9. 在「AI」Tab中，你可以見到這3個變數：learning rate(學習率), refresh factor(刷新因子) and y (gamma) exploration。



10. 探索參數代表探索的頻率：數值為 0 - 1 ，觀察它對機器人行為的影響。

詳細的數學解釋可參考 : p.12-p.13

學習/訓練 - 利用相機進行避障 |LEARNING - OBSTACLE AVOIDANCE WITH CAMERA

引言 :

你已經知道機器人如何調整其連接以選擇最相關(最好)的動作，這次我們將為機器人添加一個鏡頭以讓它可以學習偵測並避開障礙物！

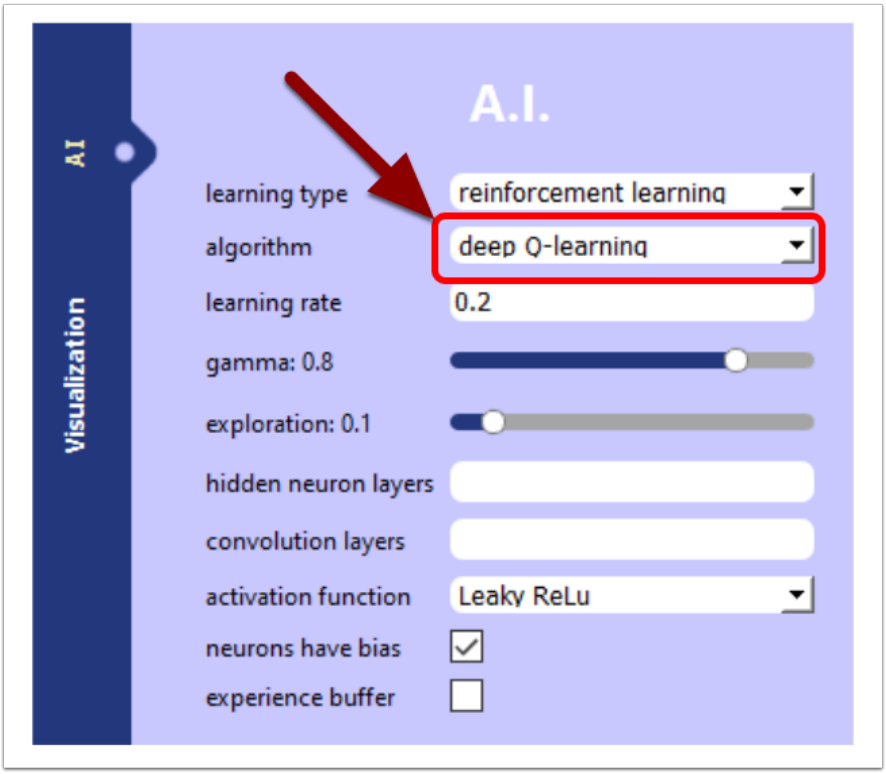
請注意，在使用相機之前，你需要重新改變新的參數！

目的：

建立正確的參數，以便在使用相機時可以有良好的學習配置。

體驗記憶 | "Experience Memory"

1. 在「AI」 tab 中，將演算法改成「deep Q-learning」，這會讓用家能使用多層的神經網絡(multi-layer neural netowrk) 並啟用新的參數。

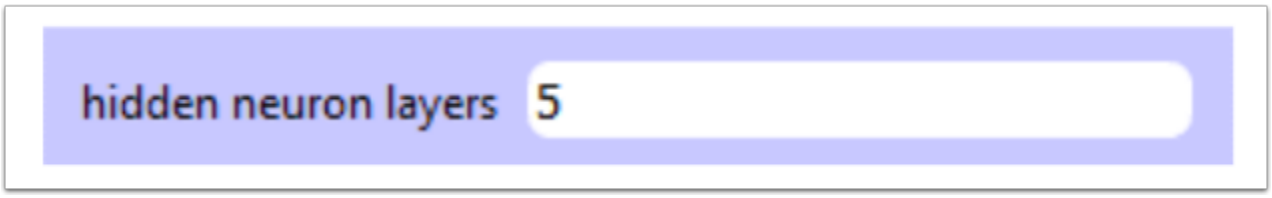


開始一個新的學習 : 你將發現連接和動作的數值在以上設定下可以是正數或負數。測試第一次的學習是否順利。

啟用「experience memory」並重置學習。一般來説它會變得更快 ! 事實上學習的速度是快速的 , 因為連接每秒都會進行修改 , 而不只是根據最後執行的操作 , 而是根據到目前為止執行的所有操作和收到的獎勵。因此 , 直接使用探索(exploration)更能學習到直行這個行動是優於轉左或轉右的。

中間神經元 | Intermediate neurons

4. 在「AI」Tab , 增加中間神經元並在界面上觀察這些新的神經元和新連接。

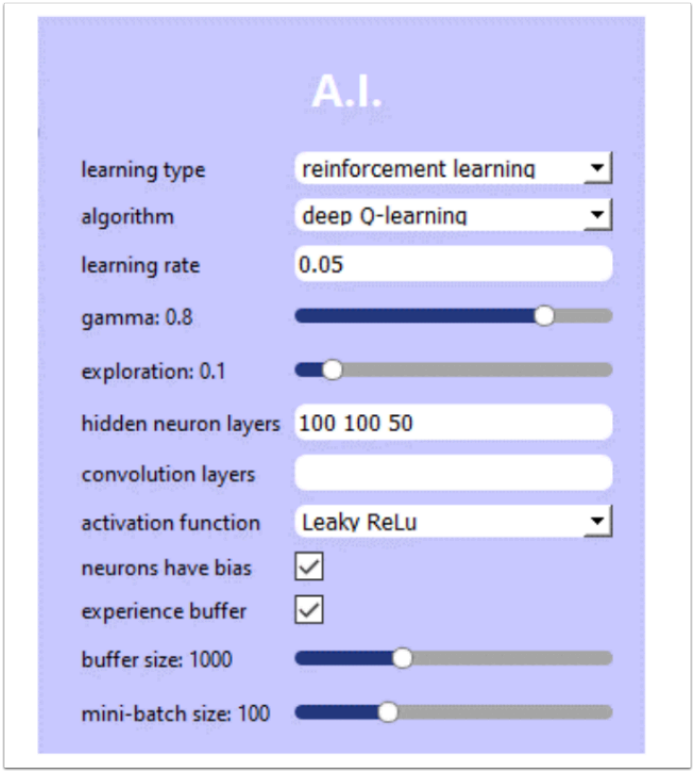


重新開始學習。和以前一樣 , 學習演算法會學習動作的正確數值 , 但計算這些數值會變得更加複雜 , 因為它需依賴大量的連接。

相機

5. **最後請在「**Parameters**」中導入「**Reinforcement Learning - Obstacle Avoidance**」**

以下是我們在「AI**」tab中的建議設定，你也能自行設定。**



6. **啟動機器人並觀察它的學習。一般來說它會依次序學習直行，然後撞到障礙物時轉彎，接著是識別障礙物, 最後是偵測到前方有障礙物時提前轉彎！本次基礎學習/訓練需時約10分鐘。你透過「**visualization**」tab 中的各個設定來查看神經網絡的連接，它的活動，它的學習過程等等...試記下你的觀察。**

學習足球 | LEARNING FOOTBALL

引言 :

你還記得機器人的目標是得到最大的獎勵嗎？如果我們更改獎勵的計算方法，我們可以讓它學習到不同的東西！

目的 :

- 訓練機器人推動綠色氣球

認出氣球 | Balloon tracking

1. 在「Demo Parameters」導入 Reinforcement Learning Balloon Tracking (green)

2. 你可以從左面的面板檢查用了什麼參數，並比較和之前的設置有什麼分別。現在獎勵設定為鼓勵機器人走向綠色物體。

3. 準備好後就在賽道/場地上擺放一個綠色氣球並開始學習。這個學習需時較長，機器人首先會學習在接近氣球時跟隨它，機器人在遠方時需更多時間學習。

4. 學習過程中，有可能沒有很好地檢測到綠色像素，在發生這種情況時，你可以：

- 加入「停止」作為可能的動作

- 把機器人放到綠色氣球前面

- 向我們查詢，我們會替你調整顏色獎勵的獎勵

5. 你成功嗎？你可以通過改變學習的設置創建更多有趣的學習！

總結 | 學到的概念 :

所謂「人工智能 | Artificial Intelligence」其實沒有一個很明確的定義，它不單只是一種方法，一個專題：再生產生物人工智能(reproducing biological intelligence)！

近年最大的進展是機器學習 : 我們已在整個 AlphAI 課程中提及過了 !

一般情況下 , 最有效的方法是使用人工神經網絡來模擬我們人類大腦中神經元的活動和學習。更準確來說 , 這是一個增加和減少人工神經元之間正確連接的問題。

就結果來說 , 某些演算法的想法其實都參考了人類自己的學習 ! 反覆試驗(Trial and error) ; 好奇心 , 學習時間 , 重複學習 , 休息等等...

其實這些演算法都是基於嚴格的數學方程式。作為人工智能研究人員必需特別擅長統計學 , 編程和基本的常識等等...