

Workshop 7 - Intruder Detection



在這一個 Workshop , 會為大家介紹 AI 神經網絡 (neuron network) 、 權重 (weight) 、 hidden layer (隱藏層) 等概念。

我們將以這個 Intruder Detection (入侵者探測器) 來演示 weight 、 hidden layer 帶來的差異。

材料：

- 已經在電腦安裝 **AlphaAI software** (每一隻 AlphaAI 需一台電腦)
- 具備藍牙或Wi-Fi功能的電腦以和 AlphaAI 進行溝通
- AlphaAI 機器人

需時：

- 30 至 45 分鐘

開機及接駁：

→ 安裝 AlphaAI software , 點擊「AlphaAI」圖示以開啟軟件

→ 開啟 AlphaAI 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。

→ 記下 AlphaI 底板的編號 (應為3或4位數 e.g 197)

→ 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)

Wi-Fi :	藍牙 :
- 將電腦連接至 AlphaI 的 Wi-Fi :找出以 ALPHA 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。

→ 點擊「connection」 button 來連接你的 AlphaI 。連接後你將在右下角能看到它的電量。

❗ 連接常見問題可查閱 : [AlphaI 連接常見問題](#)

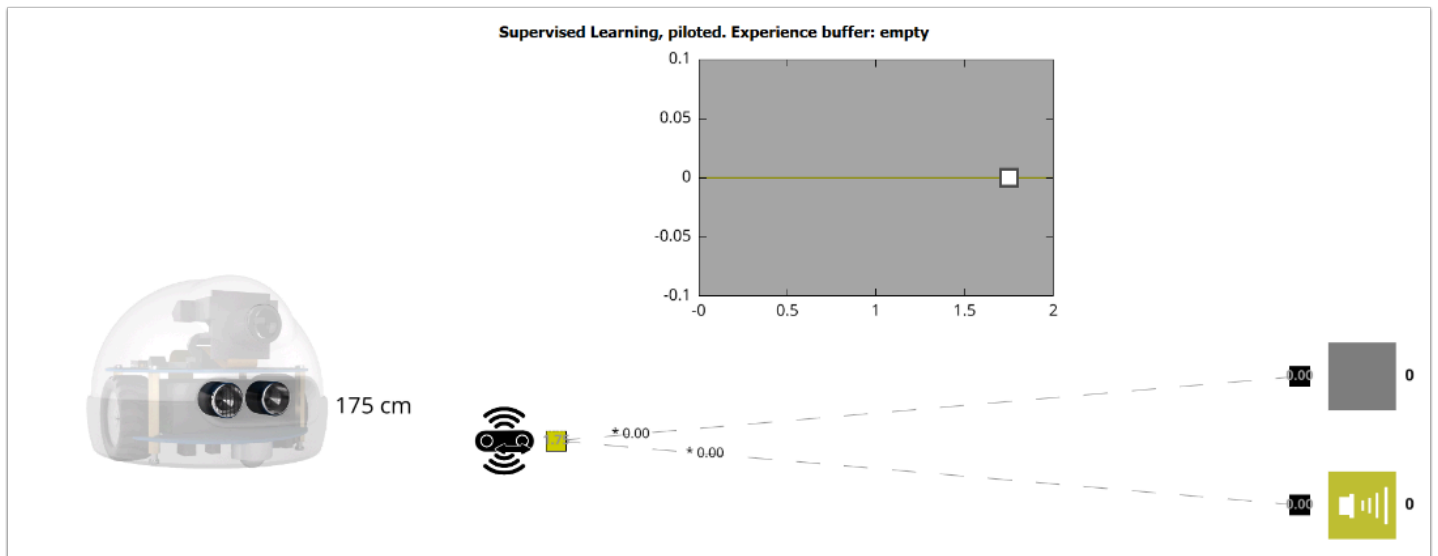
導入「Paramaters - Intruder Detection」



在「Parameters」中選出「load demo parameters...」然後導入「Supervised learning - Intruder Detction」

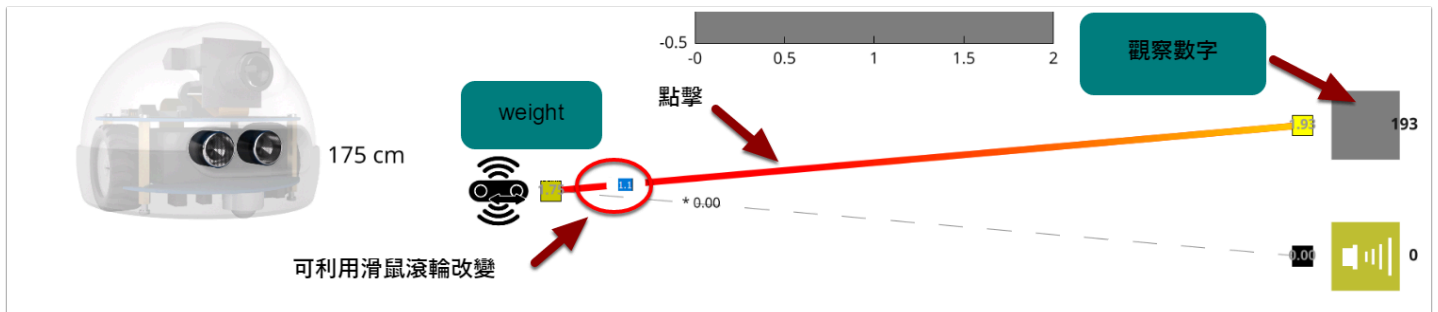
Parameters 是 AlphaI 預設的 Demo Configuration setting , 方便使用家直接導入使用。

手動編輯連接 | MANUALLY EDITING CONNECTIONS



- 透過自己創建連接(Connection)來理解簡單2層的神經網絡(2-layer neural network) [輸入及輸出 | input and output]

1. 創建與「Stop」的 Connection

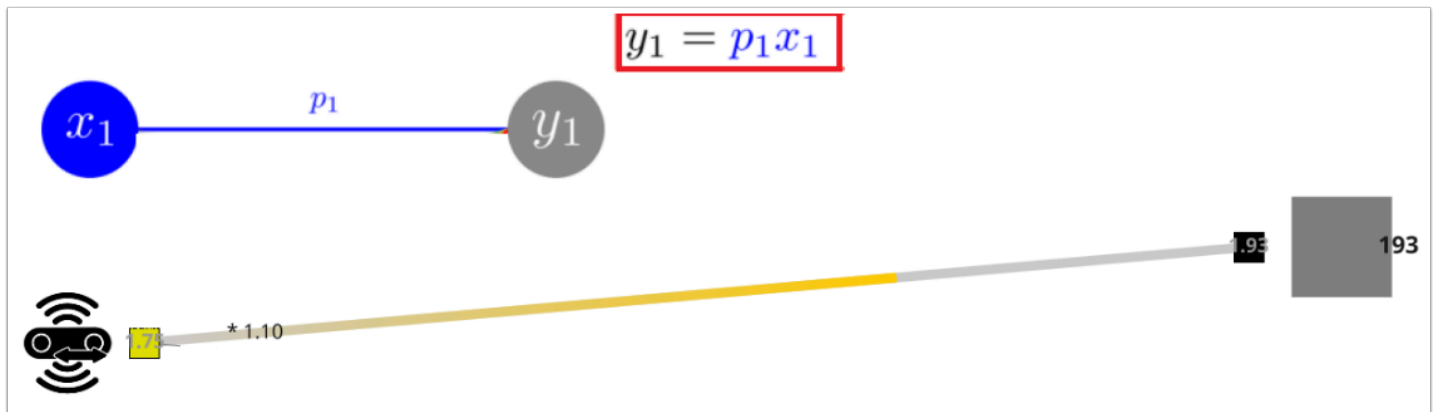


首先點擊與「stop」連住的虛線(dotted link)使它變成黃色閃爍實線，

透過改變 input 數值的 weight 看看 output 數值有什麼改變？

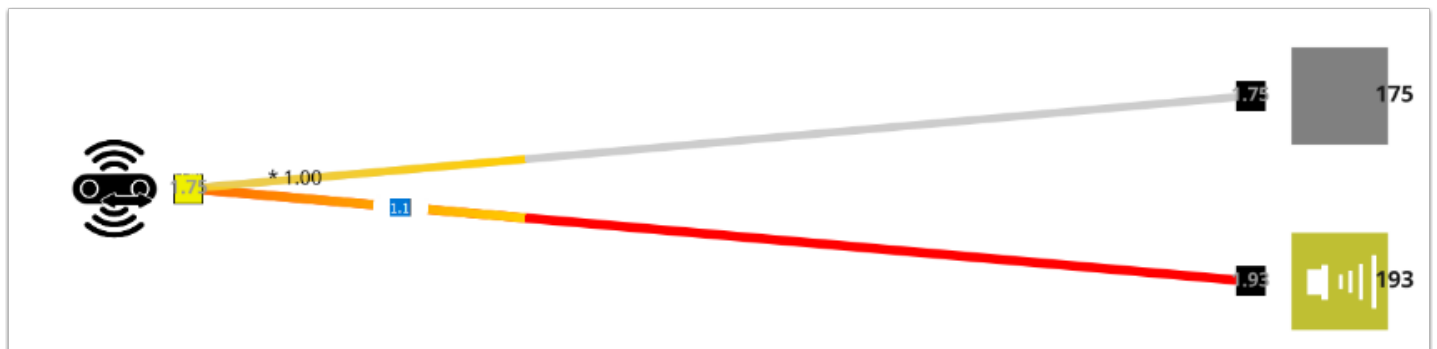
你應會發現 input 的 weight 數值愈高 Output Action 的數值也會愈高，代表更傾向執行該動作。

神經網絡：



神經網絡是一層層組成的，每一層可以有多於一個的神經元(neuron)，
在 AlphaI 的 dashboard 上的 neuron 會以一個帶有數值的正方點呈現，
而把 neuron 連在一起的數值則稱為 weight (上圖為1.1)，
Output 的數值是 Input * weight 的總和，上圖為最簡單的 1 to 1 connection，
因此 $y_1(\text{output} \mid 193) = p_1(\text{input} \mid 1.75) * x_1(\text{weight} \mid 1.10)$

2. 加入「Beep」Connection：



點擊與「beep」連住的虛線(dotted link) 使它變成黃色閃爍實線，
同時利用滑輪將 weight 的數值增加並超過「stop」的 weight，
Output 數值會有什麼改變？AlphaI 會執行什麼動作？
然後試將「stop」的 weight 改成比「stop」大，AlphaI 這次會執行那個動作？

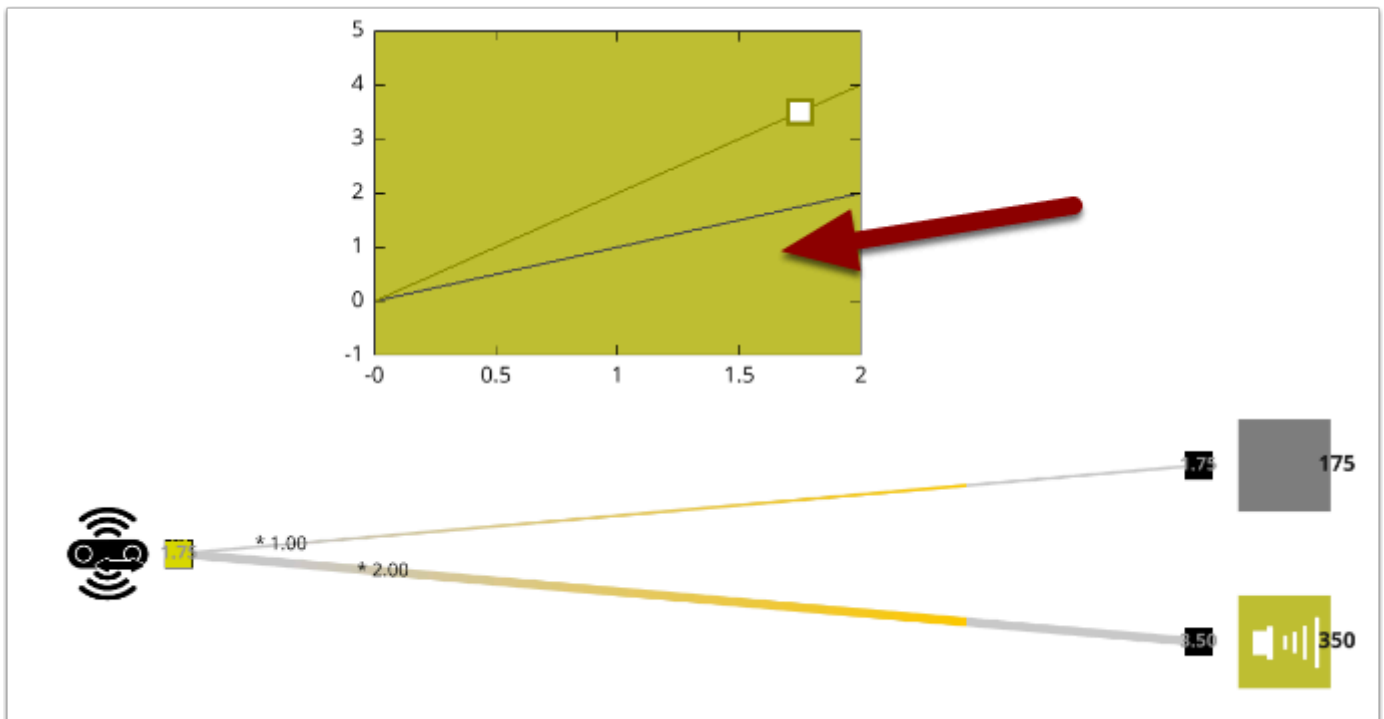
結論：AlphaI 會執行 OUTPUT VALUE 最大的行動

3. 加入入侵者



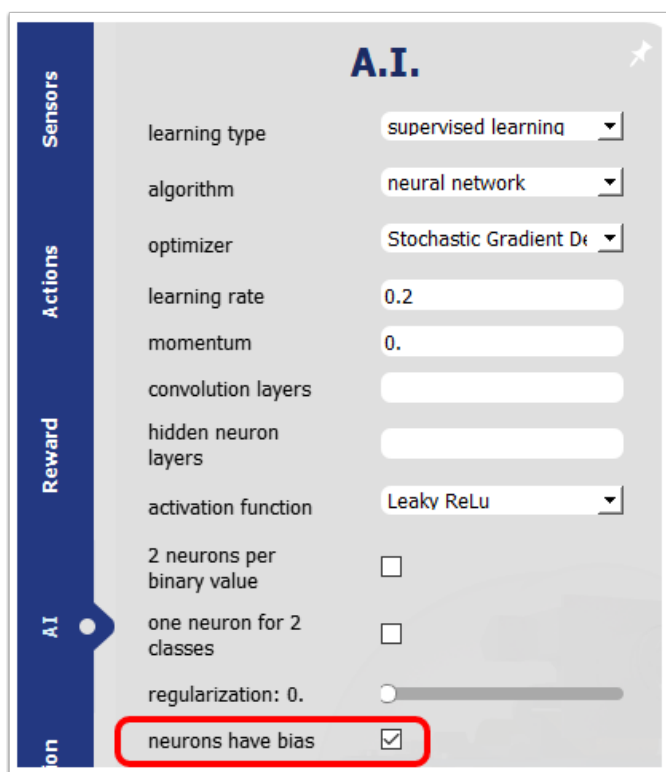
嘗試透過更改 weights , 讓機器人靠近入侵者時發出「beep」並在移離入侵者只面對牆壁時停止聲響

你會發現這是不可能的, 就算你把入侵者移走機械人都會繼續發出「beep」聲。稍後我們將會作解釋。



就上圖可見，即使你怎樣改變2個OUTPUT的WEIGHT，圖表也只會呈現一種顏色代表 AlphaAI 只會執行一種結果。

4. Neurons have bias



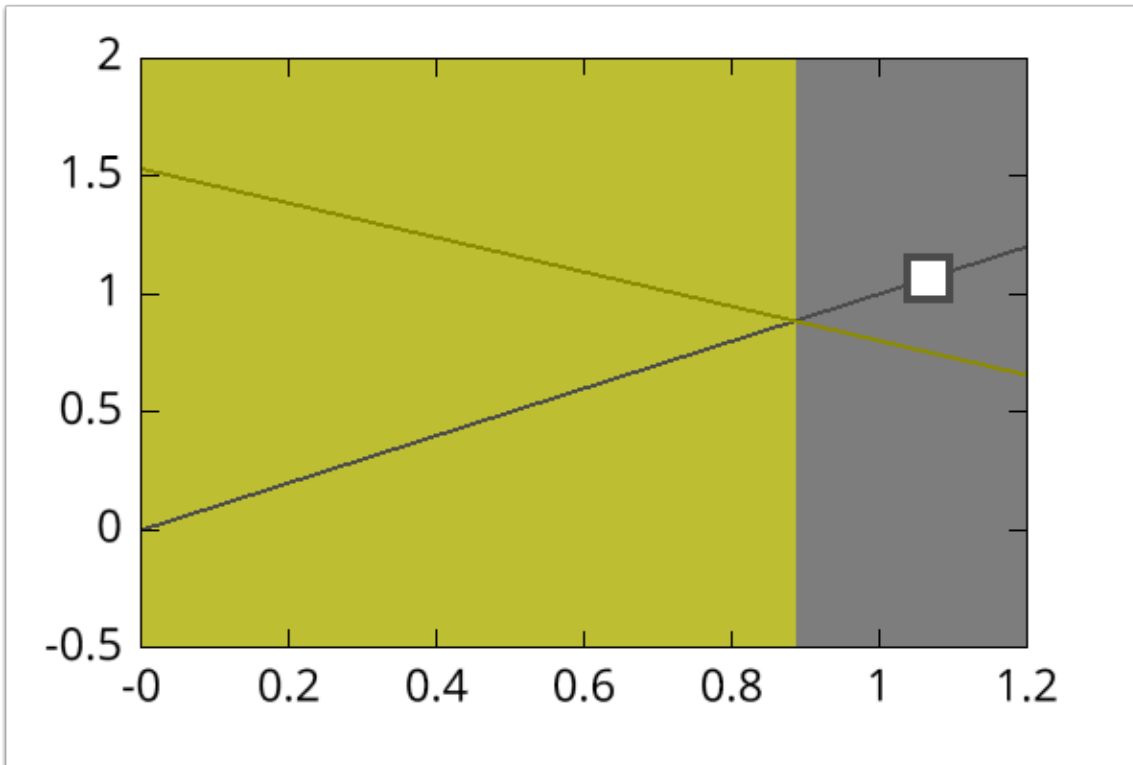
The screenshot shows the 'A.I.' configuration window. On the left, a vertical sidebar has tabs for 'Sensors', 'Actions', 'Reward', 'AI', and 'ion'. The 'AI' tab is selected. The main area contains various settings for a neural network. At the bottom, the 'neurons have bias' checkbox is checked and highlighted with a red rectangular box.

Category	Setting	Value
learning type	learning type	supervised learning
	algorithm	neural network
	optimizer	Stochastic Gradient Descent
learning rate	learning rate	0.2
	momentum	0.
	convolution layers	
hidden neuron layers	hidden neuron layers	
	activation function	Leaky ReLu
	2 neurons per binary value	<input type="checkbox"/>
one neuron for 2 classes	one neuron for 2 classes	<input type="checkbox"/>
	regularization: 0.	<input type="range"/>
	neurons have bias	<input checked="" type="checkbox"/>

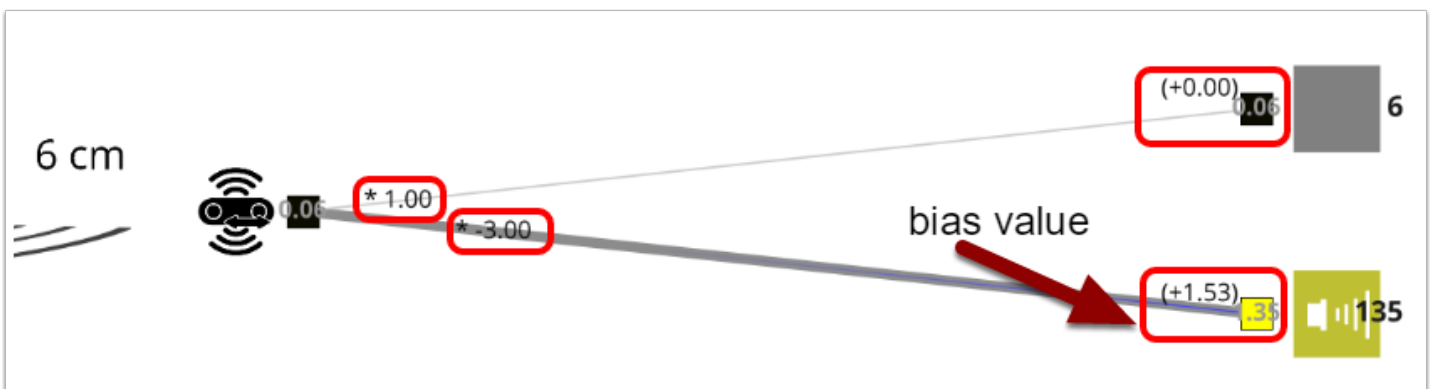
在「AI」勾選「neurons have bias」

目的是：

- 理解什麼是神經偏差
- 手動改變神經網絡來測偵入侵者
- 我們希望2條預測線可以互相交差



我們將這些不同顏色的區域稱為「決策區 | decision zone」，區間上的顏色會對應選擇動作的顏色。

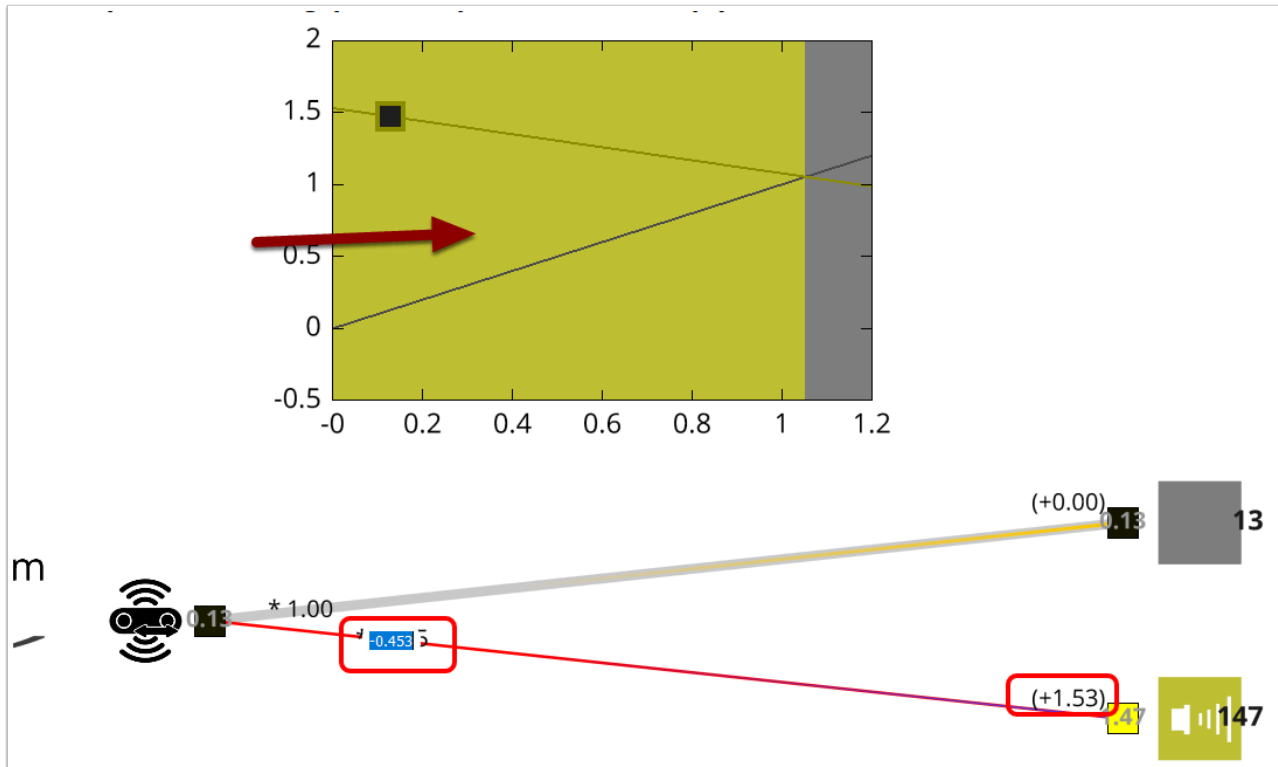


Output = Input * Weight + bias value

上圖例子 : $(0.06 * -3.00) + 1.53 = 1.35$

在 6cm 的情況下「beep」配合 -3.00 的 weight 加上 +1.53 的 bias value 得出的數值會大於沒有 bias 的「stop」

4.1 改變 weight 及加入 bias



加入 bias ,同時利用滑鼠滑輪慢慢改變 weight 的數值 (bias 數值保持不變) ,

觀察圖表上 decision zone 的變化 , weight 的正數值愈大 「beep | 黃色」應愈大 , 負數值愈大則黃色愈小。

同時嘗試做出靠近障礙物時發出「beep」聲, 障礙物消失時停止「beep」

→ 最簡單的做法是 weight 改成負數然後加入一個 bias value

總結 :

Nueral bias (神經偏差) 能令原來2條平行的預測線交差 ,

從而讓 neuron 執行其他行動。(而非原先一成不變的行動)

5. 將入侵者以 45度角擺在 AlphaAI 面前



觀察 ultrasonic sensor 顯示的數值。同時參考之前做過的預測曲線圖，為何這次機器人沒有發出「beep」聲？

6. 設定 Hidden Layer (隱藏層)

目的是：

- 透過改變 network 來偵測不同方向的障礙物
- 理解加入 hidden layer 的意義

Sensors
Actions
Reward
AI

A.I.

learning type: supervised learning

algorithm: neural network

optimizer: Stochastic Gradient Descent

learning rate: 0.2

momentum: 0.

convolution layers:

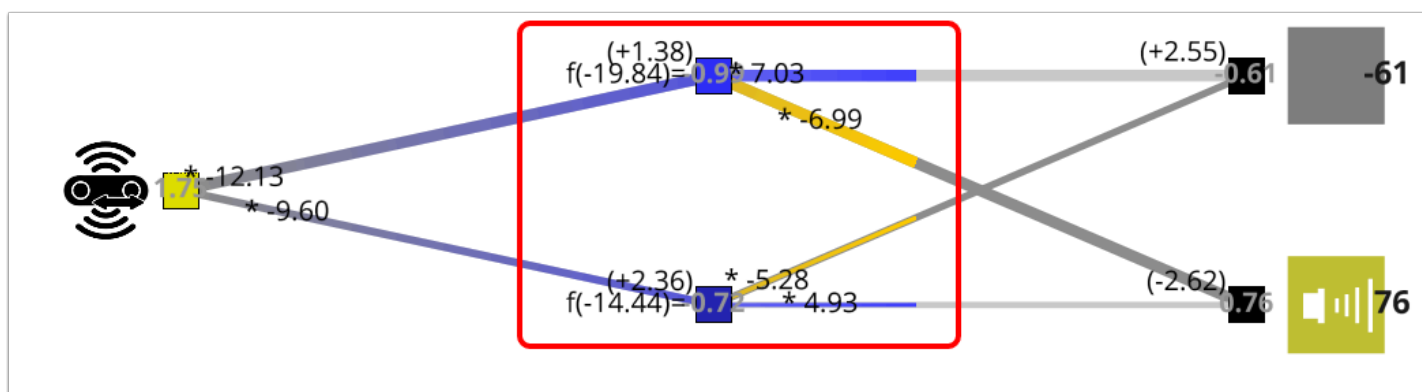
hidden neuron layers: 2

activation function: Leaky ReLu

2 neurons per binary value: ☐

one neuron for 2 classes: ☐

打開「AI」Tab , 然後在「hidden neuron layers」輸入 2



加入後 network 的中間應該會多了 2 個 neuron (hidden layer)

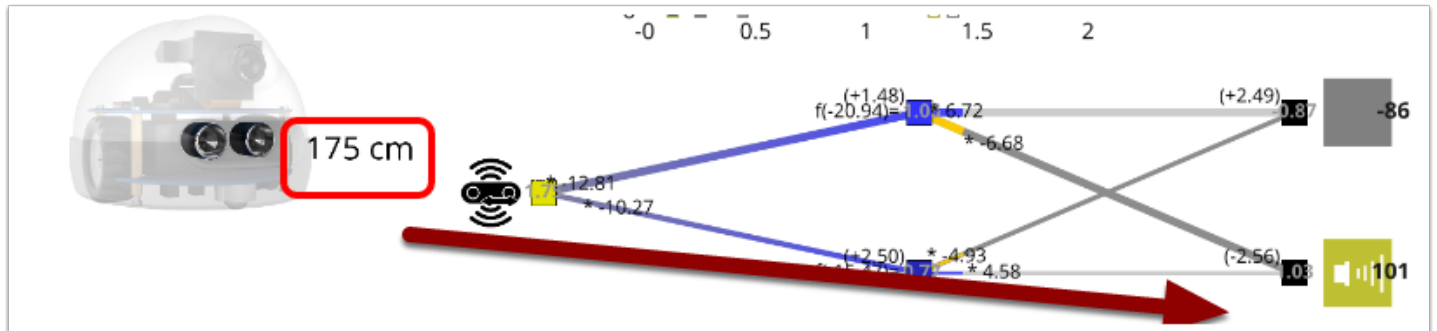
6. 訓練 AlphaI 避開 45 度角的障礙物

訓練內容分成 3 大部份：

1. AlphaI 正面偵測到障礙物 | Ultrasonic sensor 的數值很小
2. AlphaI 以 45 度偵測到障礙物 | Ultrasonic sensor 的數值很大 超過 100
3. 路上沒有障礙物 | Ultrasonic sensor 的數值大約為 40

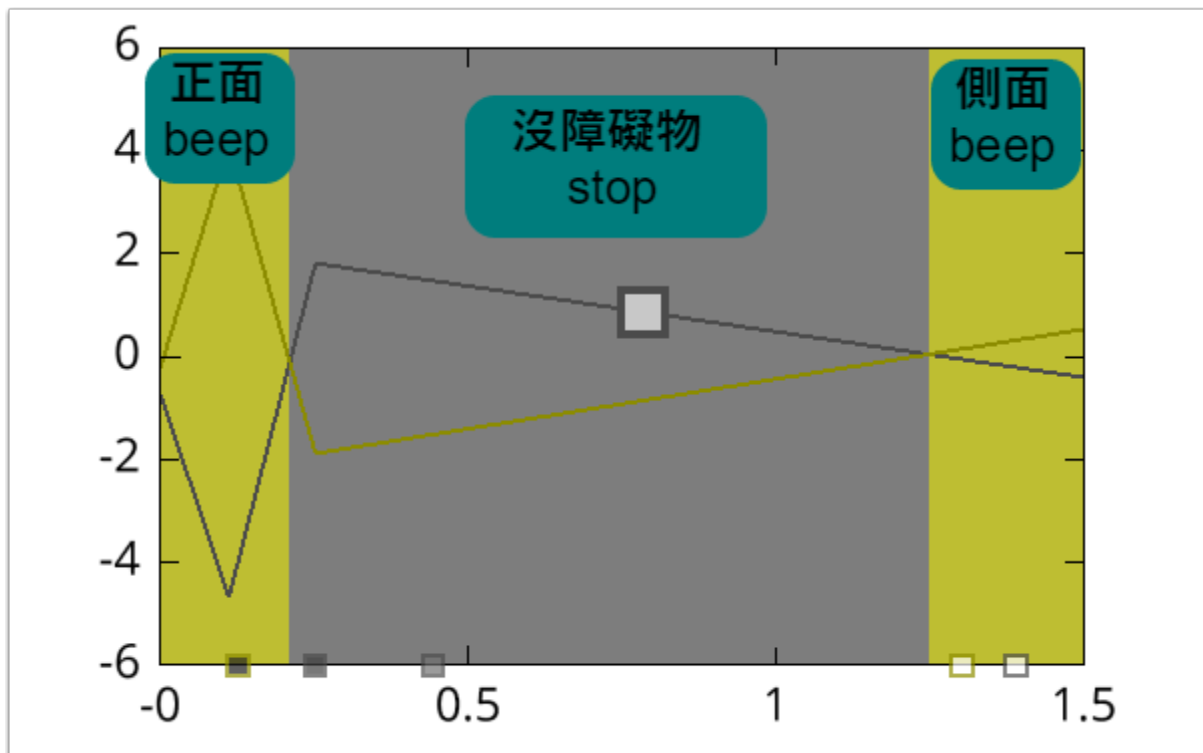


- 按下「Reset learning」
- 開啟「learning」



根據 Ultrasonic sensor 的數值來選擇 Output 進行訓練

- < 10 cm 正面有障礙物 = beep
- > 20 cm < 90 cm 沒有障礙物 = stop
- > 100 cm 側面有障礙物 = beep



你應會得出以上 3 個 decision zone (在沒有hidden layers的情況下無法做到)

添加 hidden layers 能更徹底地處理不同情況，特別是處理如以上例子般的非線性函數。