

1.

- introAI.hill.climbing.ipynb

- $f_1(x) = 2x^2 - 7x + 3$ 的最小值是多少？
- 直接執行所給的程式，程式所報告的 $f_1(x) = 2x^2 - 7x + 3$ 最小值是多少？
 - 想想看為什麼程式沒有回報正確的最小值
- 一些可能的變更
 - 加大 max_iter？
 - 如果所有參數不動，max_iter 要多大才能找到 "不錯的答案"？
 - 慢慢加大 step_size？
 - 如果所有參數不動，step_size 最大可以多大還能找到 "不錯的答案"？
 - 如果 step_size 非常大的話，程式會找到答案嗎？
 - 例如把 step_size 設定為 1

a)

$f_1(x) = 2x^2 - 7x + 3$ 的最小值：-3.125

- b) 直接執行程式所回報的最小值：20.6800000000000454，沒有給出近似最小值得原因：
 - 1. max_iter 不夠大 2. step_size 不夠大
- c) max_iter 為 3726 時，程式所給出的近似解：-3.123847999999983
- d) step_size 為 0.1 時，程式所給出的近似解：-3.120000000000007
- e) step_size 過大時，找不到答案，例如 step_size=10，程式所給出的近似解：25

2.

- simplest.case.ipynb

- 直接執行所給的程式，程式所報告的 $f_1(x) = 2x^2 - 7x + 3$ 最小值是多少？
 - 所回傳的數值是否是這一個函數的最小值？
- 如果把起始的猜測 (guess) 改為 3 的話，程式所報告的 $f_1(x) = 2x^2 - 7x + 3$ 最小值是多少？
 - 程式所回傳的數值是否還是接近函數實際最小值？
 - 這一次所回傳的數值是否跟先前令 guess = -3 的時候所回傳的數值相同？
 - 想想看為什麼會觀察到以上這樣的現象？
- 自行試試看把 guess 設定為不同數值時，程式回傳的最小值是否維持相同？或者接近？
- 如果程式其他參數都不動，把 rate 改成 0.0001 的話，程式所回報的最小值是多少？
 - 答案仍然正確嗎？
- 看懂這一個程式的所有指令，然後我們要用這一個程式來找 $f_2(x) = x^4 - 8x^2 + 3x$ 的最小值
 - 不改動原本程式的其他參數，程式回傳的最小值是多少？程式的答案是否正確？
 - 如果把 guess 改成 3、其他參數不動的話，程式回傳的最小值是多少？答案是否還是正確？
- 如果把 guess 恢復為 -3，但是把 rate 改成 0.3 的話，程式是否還能找到最小值？
 - 思考一下，為什麼這一次會有問題？

- a) 直接執行程式所回報的最小值：-3.124999999900968，這個值是近似解
- b) guess 改為 3，程式回報的近似解：-3.124999999115783，這個值是近似解，這個近似解與 guess=-3 時不完全相同，但是幾乎一樣，因為在範圍 -3 與 3 之間，這個函數並沒有其他梯度為 0 的地方
- c) 把 guess 設定為其他值，程式仍舊可以找到最小值的近似解，因為 f_1 函數只有一個 global minima
- d) 程式其他參數都不動，把 rate 改成 0.0001，程式所回報的最小值：-3.124999121371401，這個值是近似解
- e) 不改動原本程式的其他參數，程式回傳的最小值是：-22.134584543515373，這個值是近似解

- f) 把 guess 改成 3、其他參數不動，程式回傳的最小值是：-10.147923736301818，
這個值不是函數的最小值
- g) 把 guess 恢復為 -3，但是把 rate 改成 0.3，程式回傳的最小值是：inf，因為
learning rate 太大，導致每個 step 都跳太大步