

人工智慧導論 HW1 Report

賴廷瑋 F44054045

Q1: 以文字說明如何實現 P1 (Brute Force) ?

A1:

利用兩個 for 迴圈計算 x 與 y 在各自範圍內的各種組合的 loss， x 與 y 皆為整數，如此一來可以暴力找出所有 x 與 y 組合的 loss 值。最後取出所有 loss 中最小的值，則必為 loss function 的全域最小值。

Q2: P2 透過調整參數 (temperature, cooling rate...) 觀察到什麼？

A2:

1. 若將初始溫度較高，則退火過程更有可能找到全域最小值，但也不能過高，否則在退火過程有可能因為高溫而讓函數不能收斂至最小值，或者是退火過程大部分時間都不能收斂；若初始溫度較低，退火過程中更容易收斂至最小值，但找到全域的最小值之機率較低
2. 較高的 cooling rate ($T' = \text{cooling rate} * T$) 因為退火較慢，退火過程若夠久，有較大的機率找出全域的最小值；而較低的 cooling rate 因為退火較快，可以更快讓退火過程收斂到最小值，但其最小值是 global minimum 的機率較低。
3. 我的模擬退火法有加入增溫機制，將在 new cost 大於 原 cost 時以一定機率 p 回溫（無論是否接受表現較差新狀態），目的是讓退火過程更有機會找到全域最小值。而最後發現，如果 p 過高，則退火過程會很難收斂至最小值，大部分時間狀態皆跳動劇烈； p 太低的話退火過程可能會很容易收斂到區域最小值而非全域最大值。

花了大量時間調整參數後發現最佳的參數如下（退火法 7 到 8 成的機率可以找到全域最佳解）：

```
ans_list = []
for i in range(10):
    ans = simulated_annealing(init_T=10,cooling_rate=0.998, Reheat_prob=0.2, T_threshold=0.05, r
    print('epoch:{}'.format(i+1))
    ans_list.append(ans)
```

```
epoch:1 step:5486, current state: X:  0.0, Y:  30.0, current T:  0.570, current cost: -30.010
epoch:2 step:5990, current state: X: -20.0, Y:  10.0, current T:  0.378, current cost: -20.010
epoch:3 step:5543, current state: X:  0.0, Y:  30.0, current T:  0.654, current cost: -30.010
epoch:4 step:5350, current state: X:  0.0, Y:  30.0, current T:  0.609, current cost: -30.010
epoch:5 step:5031, current state: X:  0.0, Y:  30.0, current T:  0.605, current cost: -30.010
epoch:6 step:5139, current state: X:  0.0, Y:  30.0, current T:  0.557, current cost: -30.010
epoch:7 step:5249, current state: X:  0.0, Y:  30.0, current T:  0.696, current cost: -30.010
epoch:8 step:6249, current state: X: -20.0, Y:  10.0, current T:  0.332, current cost: -20.010
epoch:9 step:5405, current state: X:  0.0, Y:  30.0, current T:  0.665, current cost: -30.010
epoch:10step:5686, current state: X:  0.0, Y:  30.0, current T:  0.477, current cost: -30.010
```

```
for element in set(ans_list):
    print('{} : {} times'.format(element,ans_list.count(element)))
```

```
-30.01 : 8 times
-20.01 : 2 times
```

Q3：比較兩方法有哪些優缺點？實作過程的心得？

A3：

暴力法的優點為可以保證找出全域最小值，但可想而知如果資料量或維度增加時，缺點就是會花費大量時間找尋最佳解；而模擬退火法優點就是對於維度較高或者大量資料可以花費比較少的時間找尋最佳解，但缺點就是不能保證一定能找到全域最佳解。

實作心得

因為這次作業的 loss function input 維度較低，利用暴力法不用花太大量時間就可以找出全域最佳解（約兩分鐘），程式碼也很簡潔；模擬退火法試了非常多次參數，才找到表現較好的一組，程式碼也較複雜，不過模擬退火一次花的時間也比暴力法短（小於一分鐘）。