HW15 Report

學號：b07901039　系級：電機二　　姓名：劉知穎

1. (20%) Policy Gradient 方法

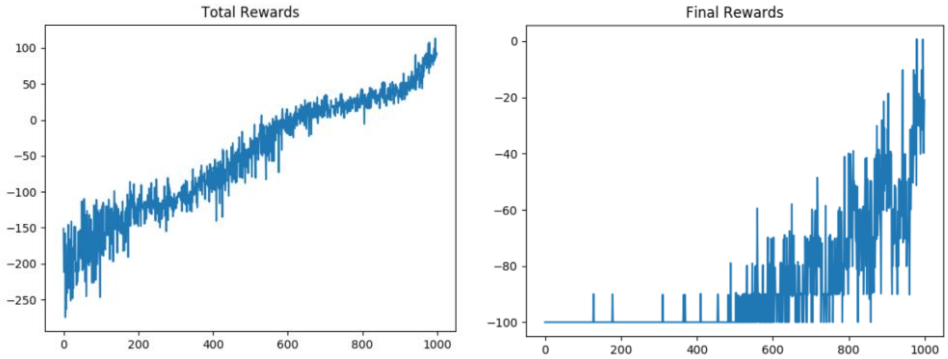
   a. 請閱讀及跑過範例程式，並試著改進 reward 計算的方式。

   b. 請說明你如何改進 reward 的算法，而不同的算法又如何影響訓練結果？

   以下訓練使用 EPISODE_PER_BATCH = 10, NUM_BATCH = 1000, 使用 Adam optimizer, learning rate = 1e-3。Policy Gradient Network 結構和 sample code 相同。Testing 時測試 200 次，以 total rewards 的平均值和標準差衡量模型的好壞。

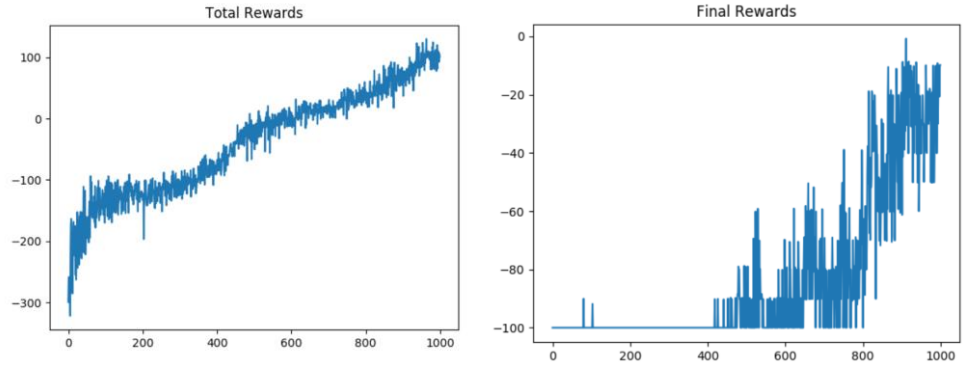
   (1) $R_t = \sum_{t=1}^{T} r_t$ (範例程式)

   Testing reward = 88.2523 $\pm$ 45.4772



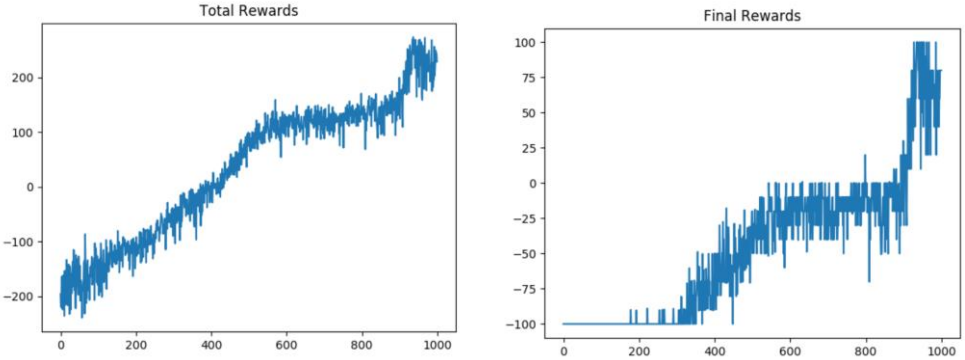
   (2) $R_t = \sum_{t'=t}^{T} \gamma^{t'-t} \times r_{t'}$

   $a.\ \gamma = 1\ (no\ discount)$



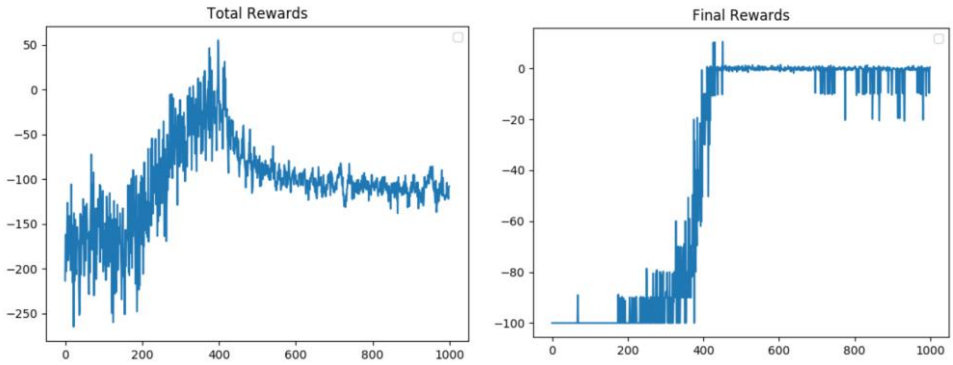
   Testing reward = 100.9911 $\pm$ 44.0909

$b.\ \gamma = 0.99$



Testing reward = 238.3641 ± 77.4247

c. $\gamma = 0.9$



Testing reward = -101.5930 ± 19.0525

結論：

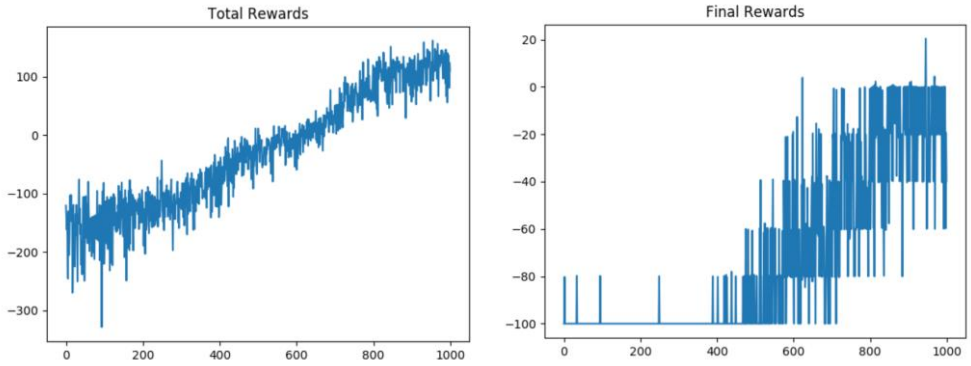
將 rewards 改成由現在的 step 累加到最後，並加上 decay 的 gamma 得到的結果較累加從頭到尾的 rewards 好。因為現在的 action 直觀上應該對之後得到的 rewards 有影響，而對之前的沒有，且影響程度應該隨時間 decay。經過實驗，$\gamma = 0.99$的效果最好。

2. (30%) 試著修改與比較至少三項超參數（神經網路大小、一個 batch 中的回合數等），並說明你觀察到什麼。

(1) 一個 batch 中的回合數等

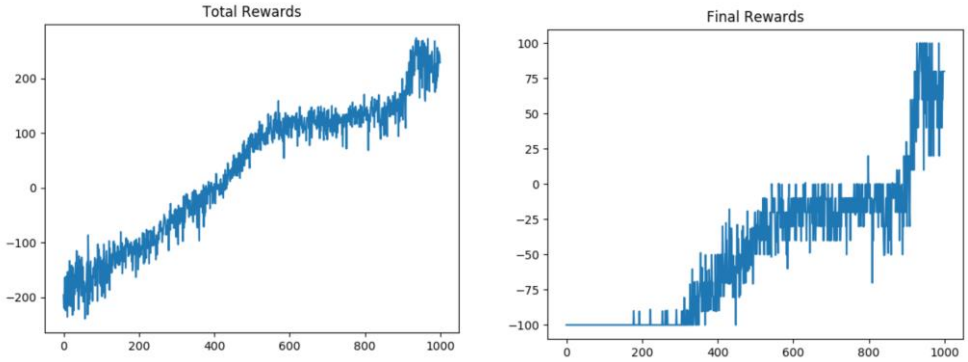
$\gamma = 0.99$, NUM_BATCH = 1000, 使用 Adam optimizer, learning rate = 1e-3。Policy Gradient Network 結構和 sample code 相同。Testing 時測試 200 次，以 total rewards 的平均值和標準差衡量模型的好壞。

a. EPISODE_PER_BATCH = 5



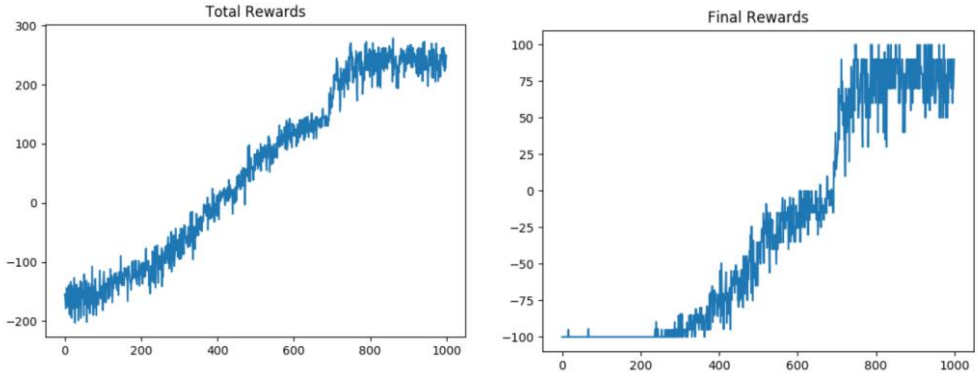
Testing reward = 107.3212 ± 58.6640

b. EPISODE_PER_BATCH = 10



Testing reward = 238.3641 ± 77.4247

c. EPISODE_PER_BATCH = 20



Testing reward = 244.7109 ± 71.1298

結論：

增加一個 batch 中的 episodes 數得到的結果會較好，因為多次 episodes 得到的 state, actions, rewards 會較全面。缺點是訓練時間會增加。

EPISODE_PER_BATCH 從 10 增加到 20 得到的 Testing reward 的成長較從 5 增加到 10 小。因此這份報告主要以 EPISODE_PER_BATCH=10 做實驗。

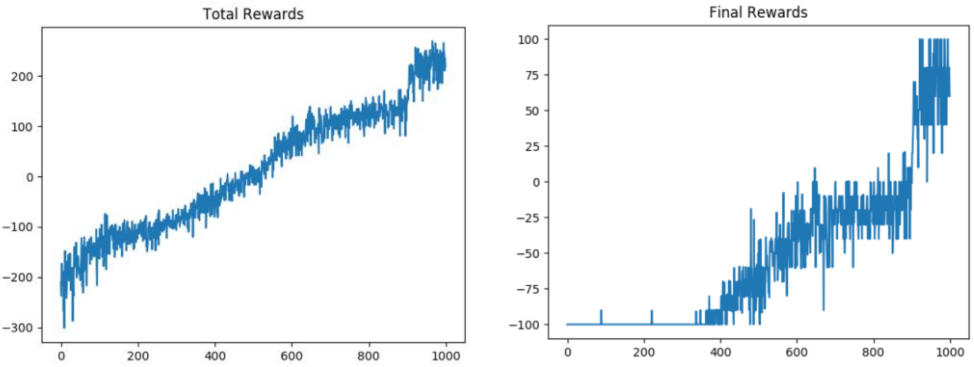
(2) 神經網路大小

Policy Gradient Network 結構：

Linear(8, hidden_size) / Tanh() / Linear(hidden_size, hidden_size) / Tanh() / Linear(hiddren_size, 4) / Softmax()

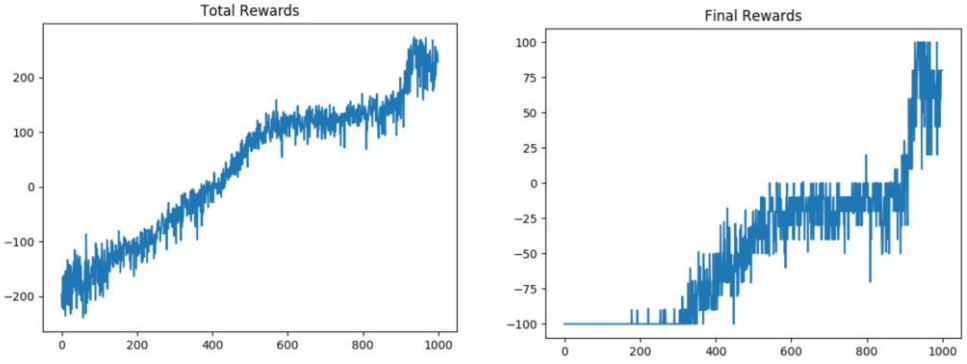
$\gamma = 0.99$, EPISODE_PER_BATCH = 10, NUM_BATCH = 1000, 使用 Adam optimizer, learning rate = 1e-3。Testing 時測試 200 次，以 total rewards 的平均值和標準差衡量模型的好壞。

a. hidden_size = 10



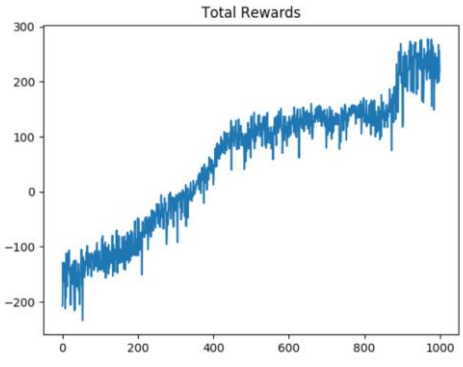
Testing reward = 234.2801 ± 78.2555

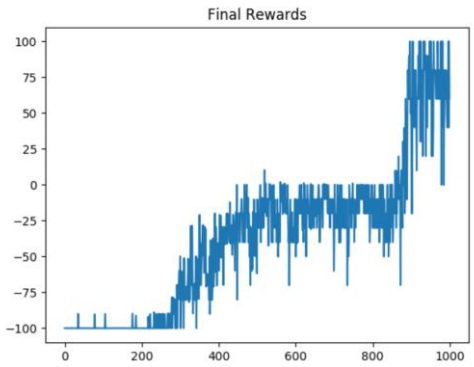
b. hidden_size = 16



Testing reward = 238.3641 ± 77.4247

c. hidden_size = 20





Testing reward = 209.1102 ± 105.6274

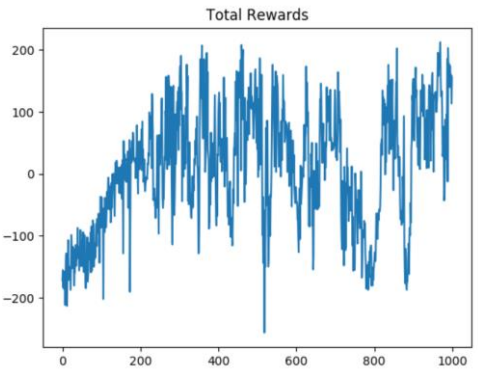
結論：

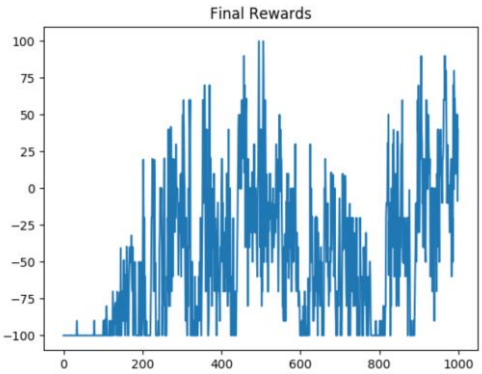
Hidden_size = 16 有最好的結果。

(3) Optimizer

$\gamma = 0.99$, EPISODE_PER_BATCH = 10, NUM_BATCH = 1000, learning rate = 1e-3。Policy Gradient Network 結構和 sample code 相同。Testing 時測試 200 次，以 total rewards 的平均值和標準差衡量模型的好壞。

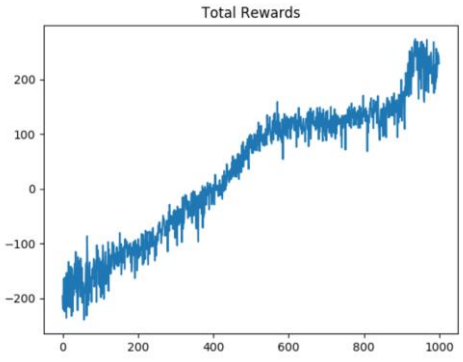
a. SGD

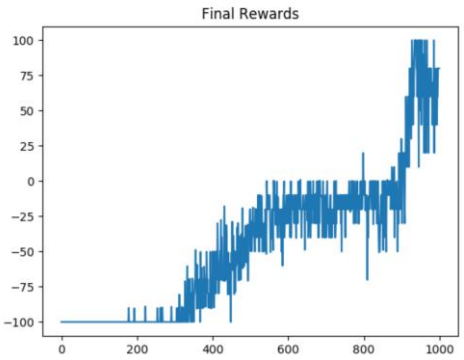




Testing reward = 139.3920 ± 68.0274

b. Adam





Testing reward = 238.3641 ± 77.4247

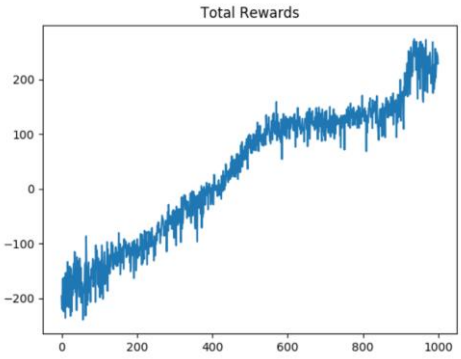
結論：

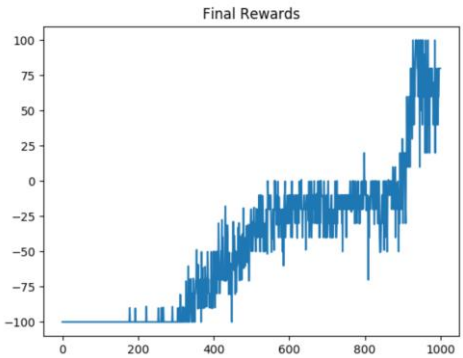
使用 Adam 訓練較 SGD 穩定且能收斂到較好的成果。

(4) Learning rate

$\gamma = 0.99$, EPISODE_PER_BATCH = 10, NUM_BATCH = 1000, 使用 Adam optimizer。Policy Gradient Network 結構和 sample code 相同。Testing 時測試 200 次，以 total rewards 的平均值和標準差衡量模型的好壞。

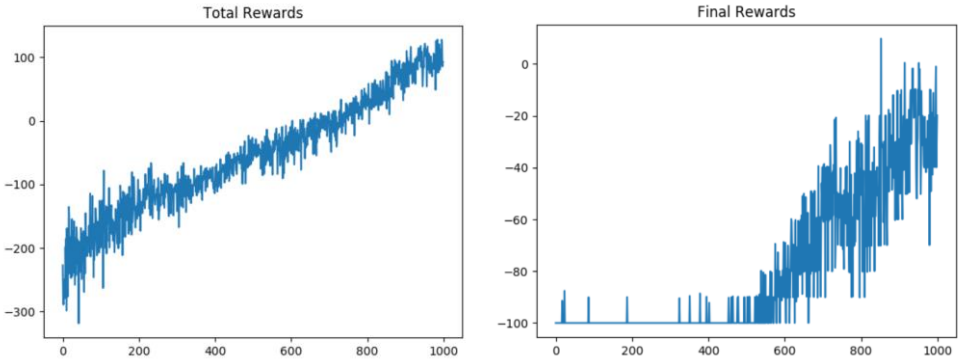
a. Learning rate = 1e-3





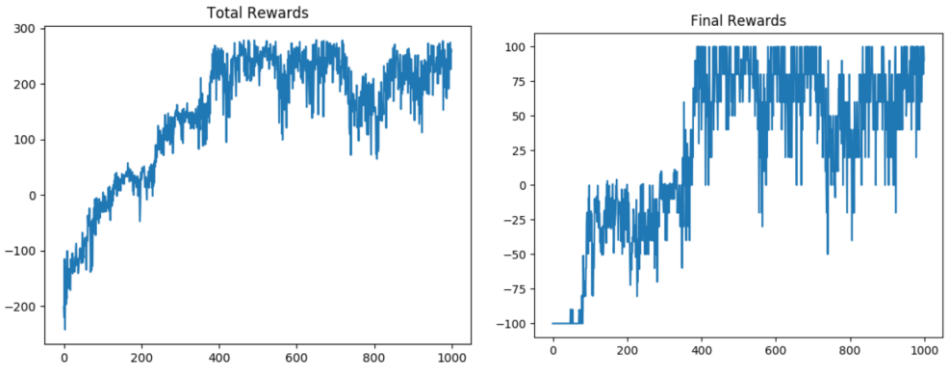
Testing reward = 238.3641 ± 77.4247

b. Learning rate = 5e-4



Testing reward = 98.1071 ± 61.4422

c. Learning rate = 5e-3



Testing reward = 255.9250 ± 51.2599

結論：

Learning rate 大時，訓練時 total rewards 成長得較快，但上下震盪也較嚴重。實驗結果 learning rate 在[1e-3, 5e-3]之間能有較好的結果。

2. (20%) Actor-Critic 方法

a. 請同學們從 REINFORCE with baseline、Q Actor-Critic、A2C 等眾多方法中擇一實作。

b. 請說明你的實做與前者（Policy Gradient）的差異。

實作 REINFORCE with baseline

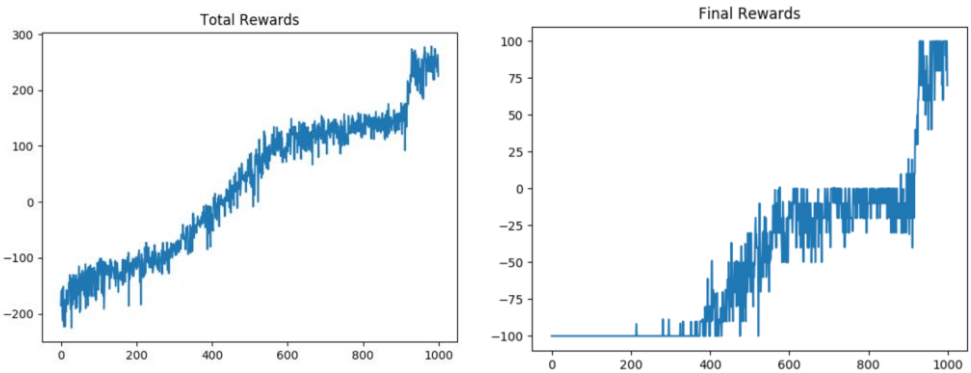
實作方法：

訓練一個小型的 linear baseline network，以一整個 episode 的 states 為 input，一整個 episodes 的 rewards 為 target。每過一個 episode 更新一次 baseline network。Advantage $A_t = R_t - b_\phi(s_t)$，以 adavantage 和 log probability 每經過一個 epoch (EPISODE_PER_BATCH 次 episodes)更新一次 Policy Gradient Network 的參數。

Baseline Network 結構：

Linear(8, 10) / relu() / Linear(10, 1)

$\gamma = 0.99$, EPISODE_PER_BATCH = 10, NUM_BATCH = 1000, 使用 Adam optimizer, learning rate=1e-3。Policy Gradient Network 結構和 sample code 相同。Testing 時測試 200 次，以 total rewards 的平均值和標準差衡量模型的好壞。



Testing reward = 251.9961 $\pm$ 45.9528

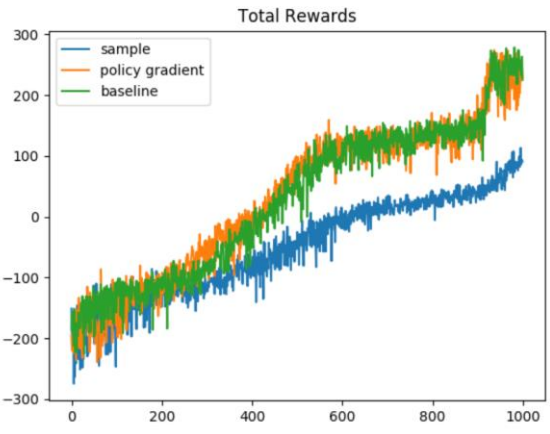
結論：

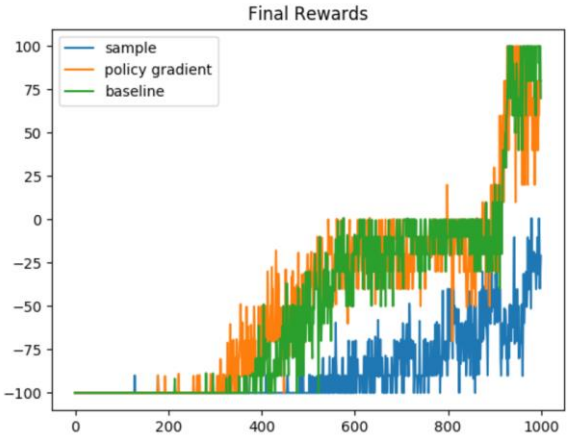
增加 baseline 後得到的 testing rewards 平均值較高也較穩定 (標準差較小)。

3. (30%) 具體比較（數據、作圖）以上幾種方法有何差異，也請說明其各自的優缺點為何。

(1) 演算法

演算法	Testing reward
Sample code	88.2523 ± 45.4772
Policy gradient + discount rewards ($\gamma = 0.99$)	238.3641 ± 77.4247
REINFORCE with baseline	251.9961 ± 45.9528



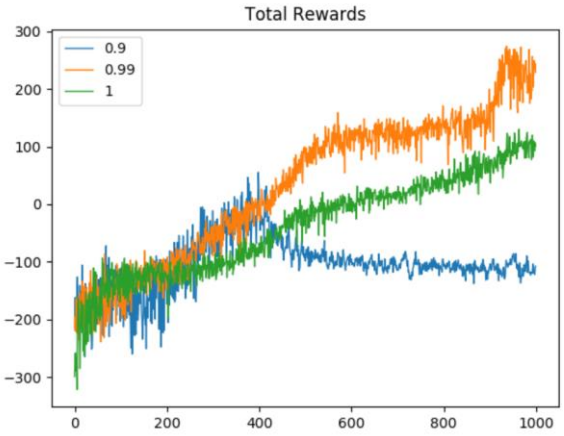


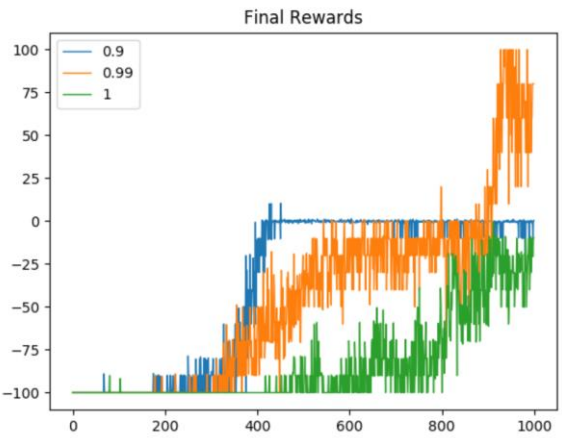
結論：

使用 REINFORCE with baseline 較原本 Policy gradient 在訓練時和測試時都較穩定，且 testing rewards 的平均值稍高於 policy gradient 的；但缺點是需要額外再訓練一個小 network，訓練上要花較多時間。

(2) 超參數

a. Reward discount γ





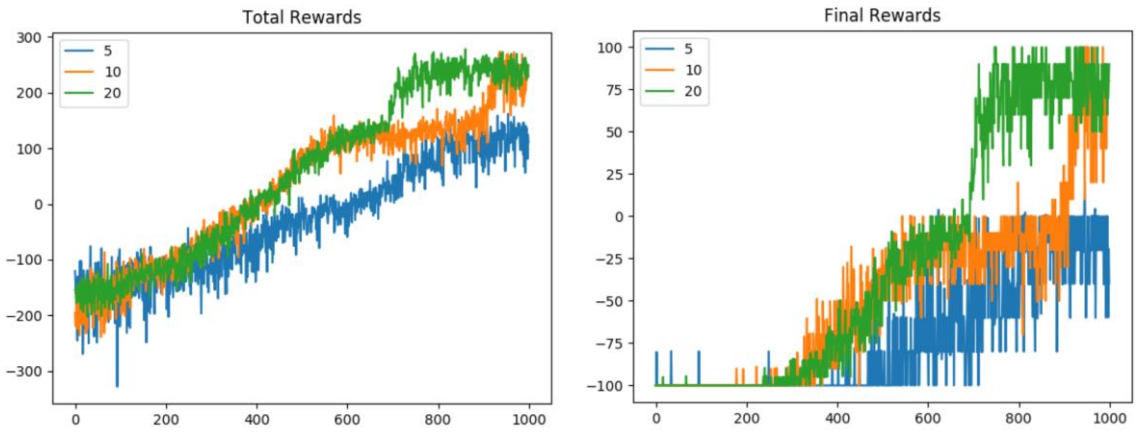
Reward discount γ	Testing reward
1	100.9911 ± 44.0909
0.99	238.3641 ± 77.4247
0.9	-101.5930 ± 19.0525

結論：

$\gamma = 0.99$的效果最好。

b. EPISODE_PER_BATCH

EPISODE_PER_BATCH	Testing reward
5	107.3212 ± 58.6640
10	238.3641 ± 77.4247
20	244.7109 ± 71.1298

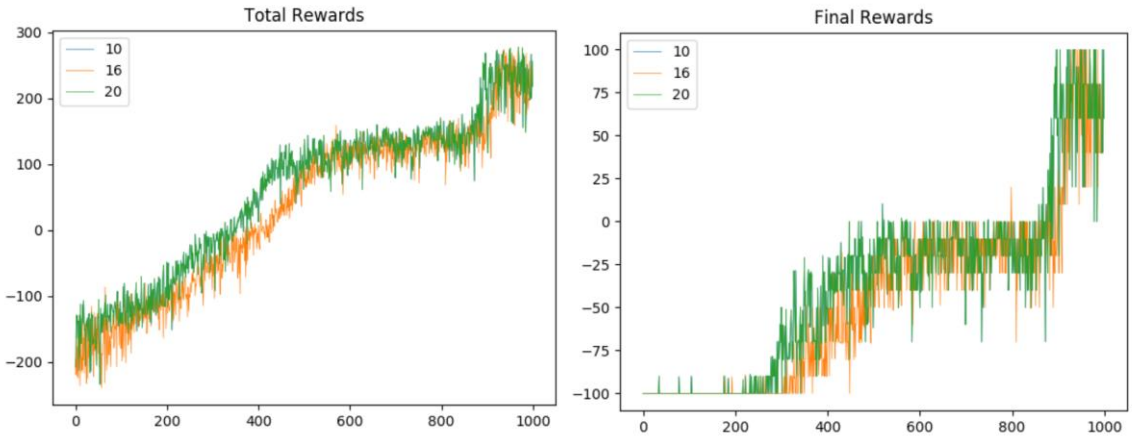


結論：

使用較大的 EPISODE_PER_BATCH 可以用較少 epoch 訓練到較高的 rewards，且最終 testing rewards 會較高；但缺點是訓練時間會增長。

c. Policy Gradient Network hidden_size

hidden_size	Testing reward
10	234.2801 ± 78.2555
16	238.3641 ± 77.4247
20	209.1102 ± 105.6274

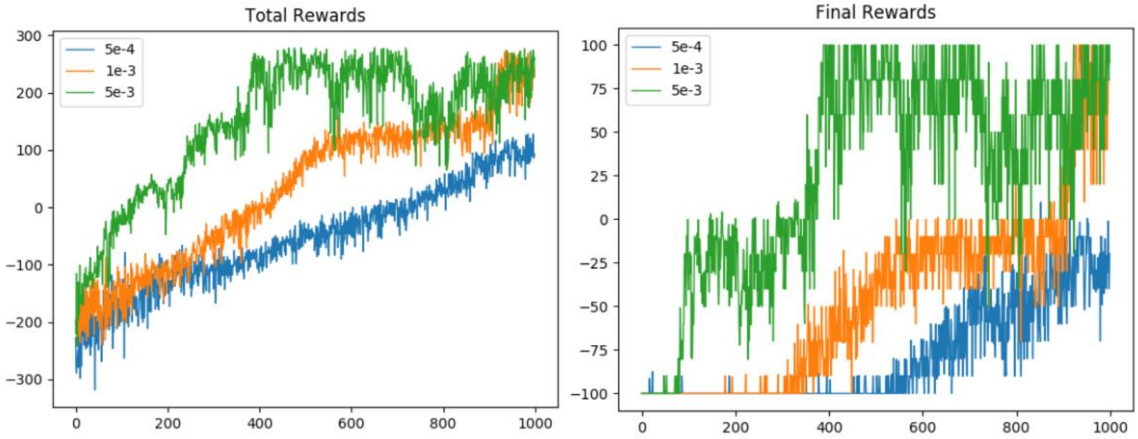


結論：

Hidden_size = 16 有最好的結果。

d. Learning rate

Learning rate	Testing reward
5e-4	98.1071 ± 61.4422
1e-3	238.3641 ± 77.4247
5e-3	255.9250 ± 51.2599



結論：

Learning rate 越大能越快到達較大的 accuracy，但訓練時 total rewards 的震盪會越大。