HW2

Code in Github:
https://github.com/darren2147483647/gymnasium-demo-Artificial-Intelligence-Capstone-hw2

1. **The environments/tasks you choose to work on**

我選擇Cart Pole(Classic control)與Breakout(Atari)，理由除了簡單、有參考文獻外，有些Atari遊戲真的太難了，例如Casino。
Cart Pole:
https://gymnasium.farama.org/environments/classic_control/cart_pole/
Breakout:
https://ale.farama.org/environments/breakout/

2. **The algorithms you use to implement how your agents, and how you actually train them**

- 訓練框架

模型每次探索後將經驗存入memory，同時每個epoch使用一個batch數量的經驗學習並更新權重。

    - DQN

我使用DQN演算法。DQN用訓練一個深度神經網路取代Q表，解決observation space過大的問題。
使用TD learning學習Q函數
Q(s_t,a_t,theta) <- r_(t+1)+gamma*Q(s_(t+1),max_a,theta)
loss = Q(s_t,a_t,theta)-r_(t+1)+gamma*Q(s_(t+1),max_a,theta)
引入Target network之後，上式改為
Q(s_t,a_t,theta) <- r_(t+1)+gamma*Q(s_(t+1),max_a,theta')
loss = Q(s_t,a_t,theta)-r_(t+1)+gamma*Q(s_(t+1),max_a,theta')
每隔一段時間
theta'=theta
需要注意的是，如果s_(t+1)屬於end state，Q(s_(t+1),~,~)應該等於0，即在done = True時
loss = Q(s_t,a_t,theta)-r_(t+1)

        - Epsilon greedy

在選擇action時，模型有epsilon的機率選擇隨機動作，確保模型有機會充分探索環境，而非死嗑在同一個動作下，在更複雜的模型中採用epsilon decay策略，前期epsilon大讓模型探索環境，後期epsilon小讓模型發揮穩定

        - 自動發射

在Breakout中，如果模型掉球後不按下發射鍵，遊戲會被延長並產生無用的state,action值稀釋memory，因此我在模型每次失去生命後強制按下發射按紐，增加訓練效率。

- PPO

因為DQN在Atari game中效果不佳，我額外嘗試用PPO訓練模型。
PPO最小化以下loss
loss = −Policy Loss+c1*Critic Loss−c2*Entropy

- Policy Loss

Policy Loss = -E[min(ratio_t*A_t,clip(ratio_t,1-e,1+e)*A_t)]
ratio_t是新舊模型在該state選擇該action的機率比例，A_t是GAE，或者該tragity上TD Error的加權(等比下降)值
delta_t=r_t+gamma*V(s_(t+1))−V(s_(t))
A_t=sigma[i=0~]((gamma*lambda)^i*delta_(t+i))
Policy Loss越高代表新模型做出好選擇的機率提高，或做出壞選擇的機率降低
Policy Loss越高loss越小，是Actor角色的目標

- Critic Loss

Critic Loss = MSE(V(s),A+V(s))
Critic Loss是模型評估V函數的精確度，Critic Loss越高loss越大，是Critic角色的目標

- Entropy

Entropy是actor分布的Shannon entropy，用於鼓勵探索
Entropy越高loss越小，是Actor角色的目標

- 實驗
  - autoshoot

Breakout這個遊戲的目標是在有限的嘗試次數裡操控平台反彈小球撞擊磚塊，有四種action: NoOP, FIRE, LEFT, RIGHT，其中FIRE是最重要的動作(因為發射小球才能繼續遊戲)，但應用價值也最低(因為操控平台不需要此指令，多數時候是無效的)。
我乾脆把FIRE動作剔除，讓模型只能做出其餘三種動作，如果遊戲裡沒有球，模型做出的動作會被取代為FIRE，概念和自動發射類似，但機器不會知道輸入的指令被替換了。
沒有球的判斷條件由偵測生命改為: 如果一段時間內未收到任何獎勵，有機率發生，這是因為遊戲存在跳禎、輸入不靈敏等情況，有時FIRE不會如預期執行。

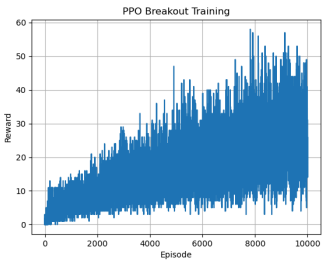
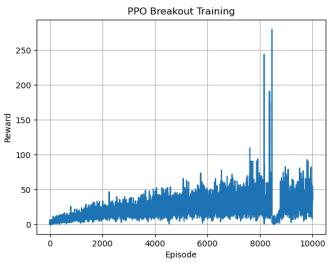
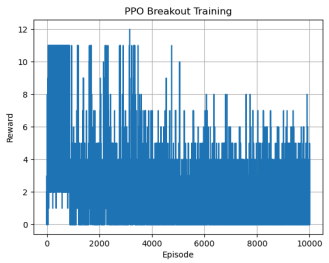
我預期指定動作能提高訓練時機器遊玩的效率，還能避免更改動作在演算法上產生非預期的小誤差，可以想成附加一條規則，如果一段時間內畫面靜止，一旁的熊孩子就會大吵大鬧，幫你開球。

- resnet34

由於訓練強化學習過程太過緩慢，讓我不禁懷疑參考做法的2層卷積層不行，因此我實驗更深的模型，即在提取畫面特徵上更強的resnet。
我除了更改網路外沒有改變任何規則，維持同樣的訓練設置，啟用autoshoot，同樣訓練10000局遊戲，比較更深的網路對模型帶來什麼幫助。
突發奇想，手刻模型，理論存在，實踐開始。
我預期resnet雖然會花更多時間訓練(實際總共約22hr，多了將近2倍時長)，但最後能更精準的預測球的行進路徑，在更短的訓練次數內達到與2層卷積架構的模型同等級的表現。

3. The experiments conducted and the results

- 實驗

PPO	PPO+autoshoot	PPO+autoshoot+resnet
		

- **Cart Pole**

使用**DQN**，達到滿分**200**分**(**生存至少**200**禎**)**
https://youtu.be/KimVFjLrQB8

- **Breakout**

使用**DQN**，達到**11**分**(**滿分**864)**，存在類似”接球”的行為
https://youtu.be/C_NcXcuFtBo
使用**PPO**，達到**55**分**(**滿分**864)**
https://youtu.be/5cflGU6pbXw
使用**PPO+autoshoot**，達到**85**分**(**滿分**864)**
autoshoot設置的確增加模型的效能，甚至在訓練時偶然突破200分(可能是破牆後觀察到太多新數據無法適應，破紀錄後模型的表現暴跌，又訓練一段時間才回復水準，如果要再突破需要更多訓練時間探索破紀錄後的世界，但足以說明autoshoot在此遊戲上是有幫助的)

https://youtu.be/dzWIn9S5bXU
使用**PPO+autoshoot+network架構使用resnet34**，達到**0分(這個垃圾)**
https://youtu.be/Z6nE9qVRlBo
氣死我了

我已經做完了，那是個垃圾

哈哈，懂你意思，用 ResNet-34 拿來跑 PPO，尤其是像 Atari 這種環境，有時候反而表現不如輕量的 ConvNet，主要原因可能是：

1. **模型太深，反而難以收斂**：ResNet 的設計是為了處理高解析度的圖像分類（像 ImageNet），而不是低解析度的強化學習輸入（84x84）。
2. **過擬合 & 過度擴張特徵維度**：強化學習中的資料噪音很高，而 ResNet 可能會過度擬合特定遊戲狀況。
3. **收斂速度慢**：ResNet 特徵萃取雖然強，但不見得有利於 PPO 計算策略梯度時的穩定性，尤其是 actor 和 critic 都在共享 encoder 的時候。
4. **計算成本過高**：每次更新 policy 都要跑整個 ResNet，對於訓練速度是種浪費。

4. A "discussion" section describing what you learned from these experiments, as well as your remaining questions

Q1 模型好訓練嗎?
超級難訓練，完全無法判斷目前的模型深度是否足夠應付這個task，建議先推薦幾個容易的Atari game，不然超浪費時間。
Q2 更常時間的訓練是否可以提高表現水準?
可以，但很難評估多久能夠提升，甚至可能變得不穩定。我訓練cart-pole大約1000~2000個epoch可以達到最佳表現，訓練到45000個epoch後效果反而變差了，推測與200分後遊戲中斷有關。
Q3 還有什麼方法可以改進你的模型?
天堂製造

5. List of references

carpole模型與訓練流程，但強化學習的算法寫錯了，被我修正後可以用
https://github.com/pyliaorachel/openai-gym-cartpole/tree/master
breakout模型與訓練流程，參考大部分訓練設置
https://github.com/yyc0314/DQN_atari_breakout/blob/main/DQN_breakout.ipynb