Squid PAIA

攻略 : RL

Outline

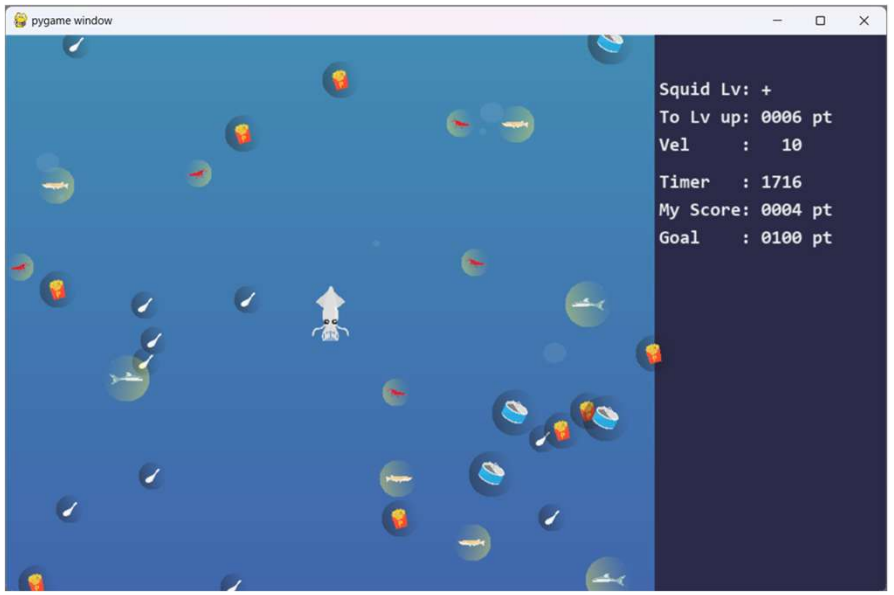
- 遊戲介紹
- 解題思路說明
  - 思路說明
    - 解題策略
    - 選擇特徵
    - 選用模型及參數
  - 實作步驟
    - 訓練模型
    - 驗證模型
  - 結果分析
    - 結果比較
    - 可能的錯誤
  - FAQ
    - 表現不如預期的可能原因
    - 我們還有其他改良的方法嗎?

遊戲介紹

遊戲介紹 - 遊戲目標

簡單的吃目標物與閃避非目標物的遊戲，遊戲中的目標物為海裡的生物，非目標物為海中廢棄物，二者皆是隨機出現，吃到目標物會加分，海廢會扣分。不同的目標物與海廢，加減分多寡會不同。如右圖。遊戲分以下兩種版本：

- 單人過關版：以在一定時間內升級到更高的層級（Level）為目標，每升高一個層級，海廢比例會增加，升級難度會變高。在一個層級裡，分數增加的同時魷魚身體也會變大，難度"可能"會增加(可以思考為何是用"可能"二字)。
- 雙人對戰版：當升到第七關後，如對手積分比自己低，可以攻擊之，反之，則需閃避。對戰雙方碰撞後會以隨機的方式彈開，此外，積分低者會扣分，積分高者會加分。比賽結束時，積分高者獲勝。

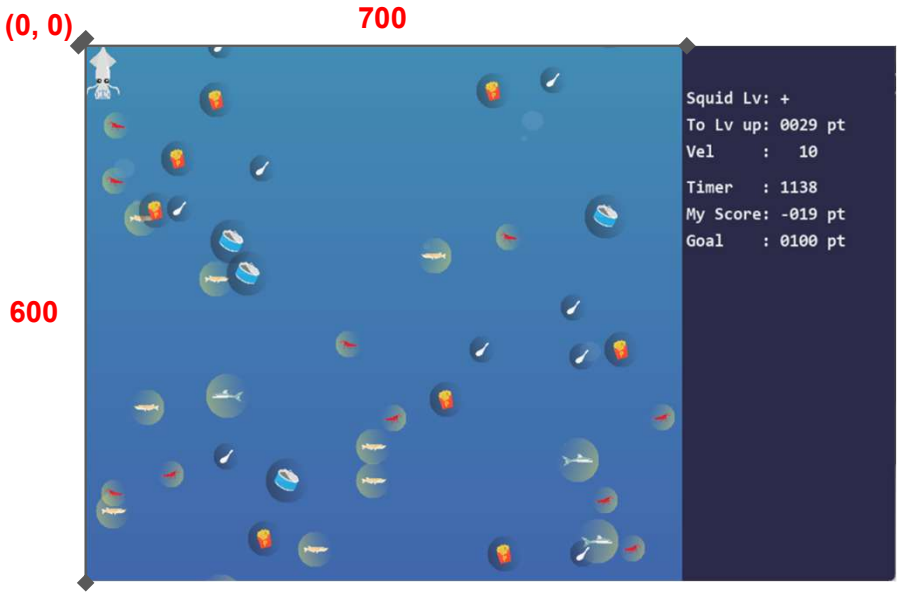


遊戲介紹 - 環境介紹

- **遊戲畫面**
  - 畫面會隨著過關層級升高而變大
- **遊戲物件**
  - 魷魚在每一個關卡中隨著得分變大，並無一定的尺寸。
  - 魷魚的活動方式分：上，下，左，右，不動，沒有斜方向移動的選項。如有移動，每次移動的速度會隨著魷魚升等而加速。
  - 食物與海廢數的量與出現位置由程式隨機決定，惟層級越高，海廢數量比例會增加，如果魷魚體積又變大了，則難度變高。
  - 遊戲畫面預設為每秒30幀，也就是每1/30秒更新一次。提供的畫面資訊(會放在積木選單)為目前關卡資訊(層級，得分等等)及畫面中每一個物件（雙人對戰版會提供對手的座標與他現在的積分）的座標。
  - 遊戲中有三種食物，分別加(1,2,4)分；有三種海廢，分別扣(1,4,10)分。

遊戲介紹 - 環境介紹

- **遊戲的座標系統**
  - 「視窗」左上角為(0, 0)，x軸向右遞增，y軸向下遞增。
    遊戲提供的座標資料都是其中每一個物件正中央的座標。
  - 但是在前期關卡中，遊戲場地並沒有擴展到整個視窗，所以場地的左上角並不一定是(0,0)。
  - 要注意因為每個物件的座標是物件正中間的座標，所以當魷魚在左上角時，魷魚的座標會是(長/2, 寬/2)。



遊戲介紹 - 環境介紹

- **表現**
  - 目前所屬層級
  - 目前還需多少分數才能升級
  - 目前移動速度
  - 剩下的時間
  - 目前總積分
  - 本層級需要的過關分數

解題思路說明

解題策略

- 解題策略
  - 本策略是以單人過關版為主
  - 採用強化式學習方法 (參考文章)
    - 強化式模型的三大要素是
      - 「觀察」（Observation）：類似監督式學習的特徵
      - 「獎勵」（Reward）：針對環境與玩家動作的好壞給予不同程度的獎懲
      - 「動作」（Action）：玩家從模型輸出的動作
  - 此次採用的是單一個模型，以魷魚本身的為中心點，分上下左右等四個區域（每個區域的範圍大小為可以作為實驗的參數)
    - 動作：因為只分四個區域，所以只有上下左右四種選擇。
    - 可能的獎勵的兩個例子：
      - 吃到越高分食物加越多分，吃到越高分垃圾扣越多分。
      - 輸出的動作是往食物越多的區域移動加越多分，往垃圾越多的區域移動扣越多分

系統複雜度

- 動作種類越多，系統複雜度越高
- 特徵種類越多，系統複雜度越高
  - 舉例來說，可以把畫面裡所有的物件（食物與垃圾）的分數與座標當作特徵，但是當關卡層級越高，畫面中的物件會太多。
  - 特徵太少不見得不好，重點在怎麼整理出特徵數目少，但是效果可能比較好的特徵。
- 系統越複雜，越不容易收斂，也需要更多訓練時間（可能數十小時）
- 不容易收斂往往代表效果不好
- 需要在效果與複雜度之間取捨

選擇特徵（也就是Observation）與獎勵方案：方法一

- **簡單，直覺**
- **前處理**
  a. 計算所有物件與魷魚的距離，排序後，選擇前N個靠近的物件，將這些的物件的座標減去魷魚的座標（與魷魚的座標差），此座標差可以四捨五入為整數，甚至進一步量化（Quantized）。
- **特徵的計算方法很多，試舉兩種**：
  1. 每個物件的特徵為物件的加減分，以及距離魷魚的座標差。N個物件排成一個特徵向量（可以比較不同的N的效果）。
  2. 既然魷魚只有四個可能的動作方向，
     a. 將物件與魷魚的座標差改為物件是在魷魚的上下左右的方位就好
     b. 將特徵重新依照上下左右分類（也可以不這麼做，可以比較）後排列
- **試舉例獎勵方案**
  a. 目前所得的分數扣掉上一個時間所得的分數（所以需要暫存上一個時間所得分數）。

選擇特徵與獎勵方案：方法二（比較複雜，但可能比較好）

- **前處理**
  a. 計算所有物件與魷魚的距離，排序後，選擇前N個靠近的物件，將每個被選取的物件的座標減去魷魚的座標（與魷魚的座標差）。根據其座標差分為上下左右四類。
- **簡化特徵並降低維度**：
  a. 將屬於每一個方向的物件的分數加總
     - 進階的做法可以將物件分數依據離魷魚的距離給予權重，越靠近的給予越高的權重，越遠離的給予較低的權重，也就是（分數）X（1/距離）。
  b. 特徵向量只剩下四個維度，每一個維度只有一個值（前述的方式，每一個特徵有一個分數與一個方向或是距離差），可表示為W(a)，其中a為上，下，左，右。
- **試舉例獎勵方案**
  a. 目前所得的分數扣掉上一個時間所得的分數（所以需要暫存上一個時間所得分數）。
  b. 除了上述a. 的獎勵方案外，多加一項，W(a)，其意義在於如果由於往加分數高的方向移動則獎勵會增加，反之如果往減分數高的方向移動則獎勵會變少。
  c. 總獎勵= 獎勵a+ λ × 獎勵b。 當λ越大代表獎勵b越重要，反之則為獎勵a越重要。

選擇模型與設定參數

- Q Learning
  - Q Learning是強化式學習(Reinforcement Learning, RL)的基本演算法之一。
  - 我們會根據要蒐集的Feature個數(維度)及範圍(各維度的範圍)建立Q Table的維度及數字。
  - Agant會根據目前狀態(state)觀察到的特徵(Observation)，調閱Q Table取得分數最高的動作。
  - 在執行動作後，計算出回饋(reward)，並透過公式來更新Q Table
  - 更新Q Table的公式如下(來自維基百科):
    - 公式中的α(學習率)及γ(衰減係數)是常數，需事先定義，可以在練習中變化

$$Q^{new}(s_t, a_t) \leftarrow (1-\alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} \right)}^{\text{learned value}}$$

其中 r_t 代表從狀態 s_t 到狀態 s_{t+1} 所得到的獎勵值, α 為學習率$(0 < \alpha \leq 1)$。γ 為衰減系數$(0 \leq \gamma \leq 1)$，當 γ 數值越大時，智慧型代理人便更加重視未來獲得的長期獎勵， γ 數值越小時，智慧代理人便更加短視近利，只在乎目前可獲得的獎勵。

參考資料: 文章一 文章二 影片一

實作步驟

訓練模型

- 與監督式學習不同，強化式學習不用事先收集資料，但也因此需要非常多倍的訓練時間。
- 訓練上幾個需要注意的地方
  - Exploration ratio(探索比例(學習率)，0到1之間的小數)：一開始可以稍微大一點，然後逐漸降低到0.05甚至更低。強化式學習如果學到一個不錯的模式往往會無法學得到新模式，此參數讓模型有一點點機率選擇到隨機產生的action。
  - 一開始的遊戲關卡沒有垃圾或是垃圾很少，會讓之後出現垃圾時難以學會此一狀況。
  - 要仔細挑選及簡化特徵，若維度太多，可能會因為記憶體原因開不了QTable。

訓練模型

訓練過程如下:

1. 建立QTable
2. 從scene_info中取得特徵
3. 根據特徵從QTable中取得最高分數的動作
    - 若多個動作分數相同則隨機選取
4. 回傳動作
5. 根據下一幀的scene_info計算reward，並更新至QTable中
6. 回到第二步驟，直到訓練結束(要注意若遊戲結束QTable會被重新建立)
7. 將QTable作為模型儲存

透過模型遊玩遊戲

流程如下:

1. 讀入上張投影片中訓練出的QTable
2. 在ml_play_model中，根據該幀的scene_info取得特徵
3. 透過特徵從QTable中取得最高分數的動作，並回傳該動作。

FAQ

表現不如預期的原因

- 計算每一等分的總分時發生錯誤
- 寫入特徵向量時發生錯誤
- 決策行動時發生錯誤
- 資料集為正確寫入檔案
- 參數設定不當或是錯誤
- 資料集過小
- 訓練資料收集過程中，海廢的數量比例太少，以至於模型比較不會閃躲海廢（因為前面的關卡中海廢確實比較少）
- 層級越高，海廢數量比例越高，理論上不可能無限打上去（所以來對戰吧！）。

我們還有其他改良的方法嗎?

- 依照食物與魷魚的距離，海廢與魷魚的距離，給予適當權重，越靠近者乘以較大權重，遠離者則乘以較小權重。
  - 用意在影響每一等分的積分。
- 可能是最後一招，也可能沒用，因為變因太多了，請大家多做實驗
  - 換機器學習模型會有用嗎?

歡迎貢獻其他攻略