Deep Learning -強化学習編-

今回の目標

問題解決のための機械学習（強化学習）的アプローチをなんとなく理解できるようにする

強化学習って何?

強化学習は機械学習の一種

- 教師あり学習
  学習用データセットに正解がある
  ex) 画像認識・・・一番似ているものを探す
- 教師なし学習
  学習用データセットに正解がない
  ex) 点群処理・・・データに潜む構造（直線など）を見つけ出す
- 強化学習
  エージェントと環境が相互にやり取りし、収益が高くなるような行動を探索する
  時間的な概念が必要となる

登場人物 #### エージェント
方策に従って、行動を行う

- 方策
  エージェントがとる行動方針（現在の状態によってのみ行動が決定する）

環境

エージェントが行動した結果もたらされる次の状態・報酬を与える

- 報酬
  行動によって得られる利益（マイナスもありうる）

強化学習の目標

収益が最大になる方策を見つけること

- 収益 G_t

$$G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots$$

時刻tの状態S_tにおいて、将来得られる報酬の合計を最大化する行動A_tをとりたい。

割引率$\gamma(0.0 \sim 1.0)$がついている理由は、強化学習が対象とする問題に２種類あるため。

- 連続タスク
  終わりがないため、割引率がないと無限大に収益が発散してしまう
- エピソードタスク
  終わりがあるが、割引率で目先の利益を優先させる

状態価値関数

時刻tにおける収益G_tが直接的に求められない場合がほとんど

- 方策が確率的な場合（複数の行動をとる可能性がある場合）
- ある行動A_tをとっても、次の状態S_{t+1}が決定的でない場合

収益G_tを考えるためには、期待値として考える必要がある

⇒状態価値関数 $v_\pi(s) = E_\pi[G_t | S_t = s]$

状態価値関数から、無限に続く報酬の和としての定義を取り除いたもの

⇒ベルマン方程式

今回考えてみる問題

- 連続タスク
- State1からState2に遷移した時に報酬+1
- 壁にぶつかった時に報酬-1
- 割引率γ = 0.8
- 方策 π: ランダムな方策
    - 右に移動する確率 = $1/2$
    - 左に移動する確率 = $1/2$

方策評価：State1

Case1: State1 -> State1 $$ 1/2 \{ -1 + 0.8 v_{\pi}(State1) \} $$

Case2: State1 -> State2

$$1/2\{1 + 0.8v_{\pi}(State2)\}$$

ベルマン方程式State1

$$v_{\pi}(State1) = 0.4v_{\pi}(State1) + 0.4v_{\pi}(State2)$$
$$0.6v_{\pi}(State1) - 0.4v_{\pi}(State2) = 0$$

方策評価：State2

Case1: State2 -> State1 $$ 1/2 \{ 0 + 0.8 v_{\pi}(State1) \} $$

Case2: State2 -> State2

$$1/2\{-1 + 0.8 v_{\pi}(State2)\}$$

ベルマン方程式State2

$$v_{\pi}(State2) = 0.4 v_{\pi}(State1) + 0.4 v_{\pi}(State2) - 1/2$$
$$0.4 v_{\pi}(State1) - 0.6 v_{\pi}(State2) = 1/2$$

方策評価

連立方程式を解くと

$v_\pi(State1) = -1.0, v_\pi(State2) = -1.5$

ここまでで、方策πに基づいてエージェントが行動した時に、得られる収益が分かった

$\Rightarrow$ 得られた収益を各状態のベルマン方程式に代入すると、各状態の状態価値関数を最大化する行動が分かる

ベルマン最適方程式 v_*

$$v_*(State1) = max(-1 + v_*(State1), 1 + 0.8v_*(State2))$$
$$v_*(State2) = max(0.8v_*(State1), -1 + 0.8v_*(State2))$$

ソルバーを用いて、連立方程式の解を求める

$$v_*(State1) = 3.05, v_*(State2) = 2.8$$

ベルマン最適方程式の解が分かると、状態Sにおける最適な行動が分かる

状態数が増えたとき

- 動的計画法
- モンテカルロ法
- TD法

動的計画法

前の状態の状態価値関数（予測値）を使って次の状態を更新する

STEP1

$V_\pi(S_0) = 0$とする

STEP2

$$V_\pi(S_0) = p(a_1|S_0)\{r_1 + \gamma V_\pi(S_0)\}$$
$$V_\pi(S_0) = p(a_2|S_0)\{r_2 + \gamma V_\pi(S_0)\}$$

というように、ベルマン方程式に従って更新を繰り返す。

モンテカルロ法

エピソードタスクでのみ可能。(終わりがあるため)

何度も方策に従って、繰り返しサンプリングすることで状態価値関数を推定する。

TD法

動的計画法とモンテカルロ法を組み合わせたアプローチ。
逐次更新していくため、連続タスクでも可能。
ある行動価値関数（Q関数）について、サンプリングして、状態価値関数を予測していく。
TD法の代表的なアルゴリズムにQ学習がある。

行動価値関数（Q関数）

ある時刻tの状態価値関数にある行動aを行ったときの価値関数

まとめ

- 強化学習を用いることで、各時刻でとるべき行動が分かる
- 現実問題を解くには、さらに状態数が増えるため、推定しないといけないQ関数が増えてしまう

$\Rightarrow$ **深層学習を用いて、Q関数をシンプルな式で近似する必要がある。**

また機会があれば。。。。