SIGNATE

Confidential

防衛装備庁主催

第3回空戦AIチャレンジ ウェビナー

令和5年12月21日　株式会社SIGNATE

SIGNATE

Empowering Your Potential

今回の課題に対するアプローチ

SIGNATE

手法	メリット	デメリット
ルールベース	• 行動判断ロジックが明確 • 実装してから対戦シミュレーションする試行錯誤のスピードが早い	• 網羅性が低くなる • 弱点が見つけられやすい • 想定外への対応が難しい
強化学習	• 行動判断ロジックをプログラムに落とすレベルまで詳細化する必要がない • 考えつかないようなロジックを自動的に学習できる	• まともに戦えるようになるまで学習に時間がかかる • 試行錯誤のスピードが遅い • 行動判断ロジックを考えなくてよい代わりにモデル構築や報酬の設計などが難しい

✓対戦における戦闘場面などをよく理解して仮説を立て戦略を立てる
- ➢ 目的は相手の全滅。制限時間や得点の計算などを考えた時にどのような行動をとるべきかを考える
- ➢ ルールベースの中に相手の動きや誘導弾の位置を予測するなどの機械学習モデルを併用するなど
- ➢ 初期行動判断モデルを参考にしたり、ブラッシュアップしてみる

✓ログ保存機能を活用して命令に対する動き方などを分析
- ➢ 動画と合わせて確認できるとよりよいと思われる
- ➢ 初期行動判断モデルの動きを観察する

強化学習の観点でのAgentの設計

SIGNATE

分散型: 1Agentにつき1機を操作すること
中央集権型: 1Agentで陣営全体を操作すること

操作方法の種別	行動空間の広さ	メモリ消費量
分散型	広くない	多い
中央集権型	広い	少ない

- ✓ 中央集権型の場合、どの機体の行動が報酬に繋がったのが分かりにくく学習が進みにくい
- ✓ 中央集権型の場合、既に撃墜された機体への行動も出力され続けるが、これは明らかに環境に影響を及ぼさないため、lossの計算で適切に処理する必要がある
- ✓ 分散型の場合、Observationの共通部分が重複するためメモリ消費量が増える
- ✓ 分散型の場合、Agentごとの個別の報酬を導入することで地面への激突のような禁じ手を学習させやすくなる
  - ➢ 中央集権型で同じことを行おうとすると報酬をベクトル化して反映先のActionノードを区別するといったような特殊な処理が必要になる
- ✓ 分散型の場合、探索時やlossの計算時に味方の行動をどう扱うかが悩みどころとなる

強化学習における報酬設計の基本的な考え方の一例

- **勝利時の最小値 ＞ 敗北時の最大値であることが望ましい**
- **勝敗に関わらず明確に避けてほしい動き(自滅など)に対しては大きな減点を与えることも有効**
- **誘導弾は、高高度、高速度で発射するほど(お互いに)射程が延びる**
  - 高度や速度に紐づいた報酬を入れる選択肢もあり
- **墜落や場外のペナルティはなるべく機体ごと(Agentごと)に与えた方が学習しやすい**

共有したサンプルコードなどを活用して報酬を実装して試されたい

初期行動判断モデル※の攻略

SIGNATE

- **攻撃重視で撃墜を試みる**
  - 今回は当たり判定が厳しめなので、攻撃のタイミングや方向を探索
  - 今回は300[s]と制限時間が短めのため短期決戦
- **回避する場合は場外に出ないように制御する**
  - 今回は場外にいる間ペナルティが加算されていく
  - 逆に相手を四隅に追い込むという戦術が有効になる可能性がある
    - ✓東西にも場外ラインがあるため

※サンプルとして提供されているアルゴリズムで、現在ベンチマークとして参戦中

強化学習サンプル(HandyRLなど)の改善に関するヒント

SIGNATE

- **学習時のルール変更(R5_contest_sample_{M/S}.yaml)**
  - 初期配置の変更や早期打ち切りの実施など
- **対戦相手の追加(R5_contest_learning_config_{M/S}.yaml)**
  - R5RandomInitialとInitialの使い分け
  - 特定の動きに特化したルールベースモデルの追加
- **Observationそのものに対する時系列情報の付加(Agentのmake_obs)**
  - 戦闘開始からの経過時間
  - 戦闘終了までの残り時間
- **Prioritized Experience Replayの導入**
- **行動(方策関数)の工夫**
  - 無効な行動に対するマスキング
  - Auto-regressiveな方策
- **Critic側に対する本来観測できない情報の供給**

更新情報

SIGNATE

- **Pythonで作成したAgentなどの独自クラスを使用したときのメモリリークを改修した。更新を反映するにはビルドしなおす必要がある。**
  - simulator_dist/root/ASRCAISim1/common.py L6
  - simulator_dist/root/include/Factory.h L23
  - simulator_dist/root/src/Factory.cpp L23-29,203
  - simulator_dist/root/version.txt L1 (バージョン番号を1.4.0→1.4.1に)
- **なお、Factory以外の挙動やIFに影響はない。**

実装のための追加Tips

SIGNATE

- **seminar/additional_tips.mdで諸々実装するためのTipsを公開**
- 独自クラスの作成に関するTips
  - 戦況に関する主要な情報の取得方法
  - 独自のRewardクラスの作成方法
  - 独自のMatchMakerの作成・使用方法
  - 作成した独自クラスとモデルの使用方法
- シミュレータ実装上の仕様に関するTips
  - C++版のjson(nl::json)をPython側で取り扱う方法
  - 強参照と弱参照

SIGNATE

Empowering Your Potential