

物体検出を用いたロボットナビゲーションにおける Deep Q-Network の学習効率化

押鐘 悠貴[†] 尾関 智子[†]東海大学[†]

1. はじめに

従来のロボットナビゲーション方法に自己位置推定と地図構築 (SLAM) [1] がある。SLAM は精密なナビゲーションを実現する一方で、事前に正確な地図構築を必要とする。正確な地図構築には時間と労力がかかり、専門的な知識を必要とする。本研究は自己位置推定と事前地図構築なしに深層強化学習によりナビゲーションを行う。RGB 画像からゴールにある目標物を物体検出し、RGB-D カメラの深度情報を組み合わせることでゴールへのナビゲーションを実現する。また、2D-LiDAR により障害物を避けながらゴールへと向かう。シミュレータは Gazebo、ロボットは Jetson Nano Mouse を用い通信制御方法として ROS を用いる。学習には、Deep Q-Network (DQN) [2] を用い、任意のスタートからゴールまでの自律走行を学習させる。

2. 提案手法

本研究は学習に物体検出を用いた先行研究 [3] を参考に縦横 4m、高さ 50cm に囲まれたフィールドを用意し、任意の始点からゴールまでの自律走行を学習させる (図 1)。フィールド内には障害物として直径 25cm、高さ 50cm の円柱が 4 つある。また、直径 7cm の球をゴール目標物とし、球を中心に半径 30cm の範囲をゴールとする。ゴールは障害物とフィールドの壁の間に生成させる。

ロボットは Jetson Nano Mouse (株式会社アールティ) を使用する。本機は上部に Jetson Nano を搭載した小型二輪移動ロボットである。センサーは前方に赤外線センサー 4 台と RGB-D カメラである Intel Realsense D435i、上部に 2D-LiDAR である RPLiDAR A1M8 が搭載されている。シミュレータは ROS のロボットソフトウェア開発において一般的に用いられる Gazebo を用いる。本研究では、インターネットで一般公開されている Jetson Nano Mouse [4]、Intel Realsense D435i、RPLiDAR A1M8 [5] のシミュレーションモデルを 3DCG ソフトである Blender で組み合わせ作成したオリジナルのロボットモデルを用い、シミュレーション上で学習、推論を行う (図 2)。

ゴールとの距離、角度はロボット前方から得た RGB 画像と深度画像を用いて取得する。最初に、MobileNet-SSD [6] の事前学習モデルを用いて RGB 画像から目標物の座標を取得する。この際使用するモ

Efficiency Improvement of Deep Q-Network in Robot Navigation Using Object Detection

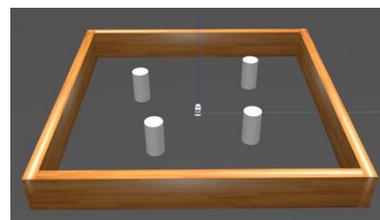
OSHIGANE Yuki[†],OZEKI Tomoko[†],[†]Tokai University

図 1. 本研究で用いる環境

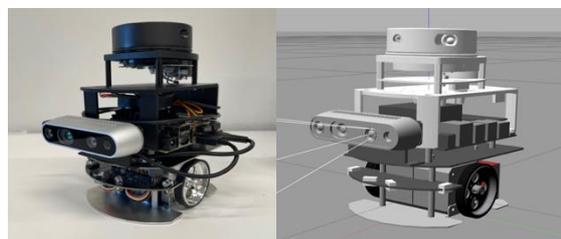


図 2. 実機ロボットとロボットモデル

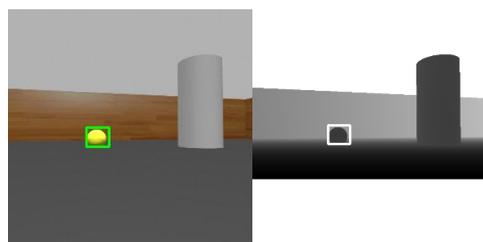


図 3. 物体検出時の RGB 画像 (左) と深度画像 (右)

デルはシミュレーションでロボットが得る RGB 画像 200 枚を用いてファインチューニングしたモデルである。次に、目標物の座標を深度画像に当てはめることによって、ゴール位置への距離と角度を算出する (図 3)。

状態は、2D-LiDAR の距離情報 (24)、最も近い障害物との距離 (1)、最も近い障害物との角度 (1)、ゴールの確信度 (1)、ゴールとの角度 (1)、ゴールとの距離 (1) の計 29 個とする。なおゴールの確信度が 0.22 以下の時は、ゴールの確信度は 0、ゴールとの角度は 0、ゴールとの距離は 5 とする。

行動は直進、2 種類の左旋回、2 種類の右旋回の計 5 つとする。直進速度は 0.1m/s、左旋回は直進成分 0.1m/s とロボットのヨー角 0.5rad/s、1.0rad/s の合成速度の 2 種類、右旋回は直進成分 0.1m/s とロボットのヨー角 -0.5rad/s、-1.0rad/s の合成速度の 2 種類である。なお、ロボットは 1 秒間に 4 ステップ行動するように均一化し、ステップ間の処理時間を等しくする。1 ステップは、ロボットが状態、行動をもとに取得した報酬、次のステップの状態を得るまでとする。

本研究で用いる DQN は状態をニューラルネットワークに入力し、行動価値を出力する。行動後の行動価値 (TD ターゲット) と現在の行動価値の差である TD 誤差を損失とし、損失を小さくするようにニューラルネットワークの重みパラメータを更新する。また、メインのネットワーク (Main Network) とは別に、同様の構造である Target Network を用い、TD ターゲットを求めることによって、更新の変動を抑え、学習を安定させる。ネットワーク構造は活性化関数 ReLU, 64 ノードの 2 つの隠れ層を持つ多層パーセプトロンである。

3. 実験方法

学習は、総ステップ数は 75,000, バッチサイズは 64, バッファサイズは 1,000,000, ネットワークの学習率は 0.0005, 割引率は 0.99, 学習エポック数は 1, 損失関数は平均二乗誤差, 最適化アルゴリズムは Adam とする。Main Network の学習は各ステップ行い、Target Network の更新頻度は 2,000 ステップごととする。また、学習中の探索率は学習開始時 1.0, 学習終了時は 0.02 まで線形的に減少させる。エピソード終了条件は衝突した時、500 ステップ経過した時とする。時刻 t での報酬 R_t は

$$R_t = \begin{cases} +10 & (Goal) \\ -10 & (Collision) \\ +5 + R_\theta \times R_d & (C > 0.22) \\ 0 & (else) \end{cases} \quad (1)$$

とする。ここで、 C は物体検出の確信度である。 R_θ はゴールがロボットの正面側の場合、正の報酬を与え、そうでない場合は負の報酬を与える ($-5 \leq R_\theta \leq 5$)。 R_d はロボットとゴールの距離が近ければ小さい報酬、遠ければ大きい報酬を与える ($1 < R_d \leq 4.34$)。学習後、学習済みのネットワークを用い 50 回推論を行い、性能評価する。

4. 実験結果

ロボットの取得したステップ終了時の即時報酬 (オレンジ線) と直近 100 ステップまでの移動平均 (黒線) を図 4 に、各ステップの損失 (緑線) と直近 100 ステップの移動平均 (黒線) を図 5 に表す。学習を重ねるごとにステップの即時報酬が増加し、

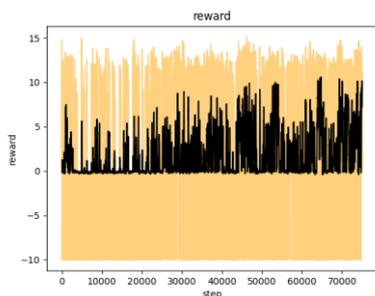


図 4. 即時報酬の推移

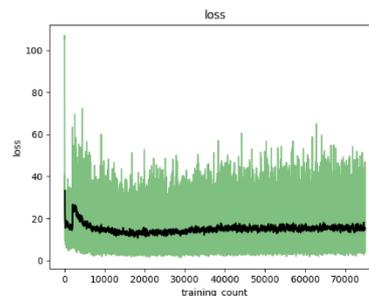


図 5. 損失の推移

学習が進んでいることを示している。また損失は 10,000 ステップまで減少し、収束した。学習後、推論実験を 2 回行い、1 回目は 50 回中 30 回、2 回目は 50 回中 32 回ゴールに到達した。

5. 考察と今後の展望

本研究では、物体検出を用いてシミュレーション学習を行い、学習後、学習済みネットワークを用い、50 回推論を行い性能評価した。結果、ゴールへの到達確率は平均 62% となった。失敗時の様子を確認すると、物体検出時に障害物とゴールが重なった際に衝突するケースがほとんどであった。一方で、ある程度遠いゴールに対しても物体検出精度は高く、長い距離をナビゲーションするケースもあった。これは事前にシミュレーションでロボットが得る RGB 画像を用いてファインチューニングしたため、距離のあるゴールに対しても物体検出精度を保つことができたと考えられる。今後の研究では、物体検出時に障害物とゴールが重なった際の報酬をより細かく設計することによって、学習精度の向上を目指す。

参考文献

- [1] H. Durrant-Whyte, T. Bailey. “Simultaneous localization and mapping: Part I”, IEEE Robot, 2006.
- [2] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wiersta. “Playing Atari with Deep Reinforcement Learning”, NIPS Deep Learning Workshop 2013.
- [3] M. R. Lee, S. H. Yusuf. “Mobile Robot Navigation Using Deep Reinforcement Learning”, MDPI, 2022.
- [4] RT Corporation, https://github.com/rt-net/jnmouse_description.
- [5] K. Klimov, <https://grabcad.com/library/rp-lidar-al-inventor-reassembly-1>.
- [6] A. Younis, L. Shixin, S. Jn, Z. Hai. “Real-time Object Detection Using Pre-Trained Deep Learning Models MobileNet-SSD”, ICCDE, 2020.