

モデルカーを用いた自動走行のための学習モデルに関する考察

千葉 笙平[†] 笹岡 久行[†]
Shohei Chiba Hisayuki Sasaoka

1. はじめに

現在,国内外で自動運転システムを搭載した自動車の開発が進み,高速道路などの一定の条件下では,公道を自動走行することが出来る. Google の自動運転車開発部門である Waymo[1]が公道で実験を行っている自動運転システムは, Carcraft という機械学習を取り入れた VR シミュレーションソフトを用いて,自動運転システムが手動運転に比べてはるかに安全であるという信頼性を確立するために,仮想空間で何十億マイルも自動走行を行って,自動運転自動車を市場に送り出そうと取り組んでいる.このように機械学習は,自動運転技術を進化させ,安全な自動運転を実現するために必要不可欠なプロセスである.

本研究は,自動運転を実現するため,また,より少ないコストにて転移学習へとつなげることを目的として, Amazon Web Service 社が中心となり研究開発を行っている, Deep Racer[2]を利用し,モデルカーの自動走行シミュレーション実験を行った.その後,シミュレーション実験の結果からモデルの性能評価・考察を行うことによりモデルの汎用性・改善策の検討を行った.

2. 提案手法

2.1 報酬関数の設定

強化学習は,価値を最大化するような行動を学習し,次に取るべき行動を予測する機械学習モデルを作成するための学習方法であり,報酬関数の設定は,強化学習によるモデルのトレーニングで重要な役割を果たす.報酬関数は,より理想的な動きを機械にしてもらうための関数であり,行動結果を評価し,それに応じて行動に報酬を与え行動の最適化を図っている.報酬関数については,“AWS DeepRacer 報酬関数の入力パラメータ”[3]にて詳細が記されている.

すべてのコースでの完走を目指すため,コースアウトしないよう,デフォルトの設定よりもセンターラインに近いほど高い報酬を与え,遠いほど低い報酬を与えた.また,カーブでコースアウトすることが多かったので,ジグザグ走行を抑制し,急ハンドルを取らないように前輪の角度を制御する,ステアリングの報酬設定を行った.

カーブをより正確に通過するために,モデルカーから最も近い2点のウェイポイントから中心線の方向を計算し,コースの方向と自身の方向の差を算出,差が大きい場合ペナルティを与え,コースの進行方向と同じ向きに進むように図った.

完走すれば多く報酬を与え,より完走率の高いモデルを目指した.

2.2 アクション・ハイパーパラメータの設定

アクションは最大速度,速度の粒度,最大ステアリング角度,およびステアリング粒度を設定でき,このアクションスペースと報酬関数の設定の組み合わせがモデルの走行制御の重要な要因となる.今回走行させる仮想コースは急なカーブが多いので,コースアウトしないよう,ステアリング角度は30度と,最大に設定し,最大速度は低めに設定した.

ハイパーパラメータ[4]は,強化学習の学習方法を設定,調整するための各種数値であり,トレーニングパフォーマンスを向上させるために使用される.今回は初期設定のまま学習を行った.

Maximum steering angle: 30 degrees

Steering angle granularity: 5

Maximum speed: 1 m/s

Speed granularity: 2

ハイパーパラメータ: 初期設定のまま

2.3 学習時間の設定

学習を停止する条件として,学習し続けることを避けるための最大学習時間(分単位で,5~1440分)を指定する.今回は前述の通り,1時間学習させたモデルと2時間学習させたモデルを比較するため,60分と120分で設定した.

3. 評価実験

3.1 実験方法

強化学習の学習処理において,走行速度,最大ステアリング角度など学習環境を統一し,コースアウトをなるべく減らすように報酬関数の入力パラメータを設定した.

全部で16コースある内の最もカーブが多く走破が難しい Fumiaki Loop のコースで,学習時間を1時間と2時間に分けてトレーニングを行い,より完走率の高いモデルが作成出来るように図った.その後,自動走行における汎用性を考察するため,いくつかの走行コースを用いてシミュレーション実験を実施し,モデルの性能評価・比較を行った.

3.2 結果・考察

表1に示すように,1時間多く学習させることにより,平均報酬,平均完走率を向上させることが出来た.3時間以上学習させることでより性能が向上する可能性があるが,2時間でも十分ピークに近い値を得ることが出来た.

表1: 平均報酬・平均完走率

モデル	平均報酬	平均完走率
1時間学習	133.40	69.00%
2時間学習	231.85	87.50%

[†] 旭川工業高等専門学校, National Institute of Technology, Asahikawa College

表2,表3から,Fumiaki Loopで走行させたシミュレーションの結果を比較すると,完走率は変わらなかったが,完走したときのタイムは,1時間学習させたモデルより2時間学習させたモデルの方が,15秒ほど縮まっていた.これは学習時間を増やしたことにより,モデルがより効率の良い走行の仕方を学習したからだと考えられる.

最初のカーブを内側寄りで曲がった際に,コースの進行方向に沿って曲がろうとハンドルを切り,コースアウトしているケースが多かったため,完走率はあまり変わらなかった.

コースのサイドにいる際,その方向にハンドルを切らないような設定を付け足すことにより,この問題は解決できると思われる.

表2 : Fumiaki Loop シミュレーション結果(1時間)

Trial	Time	Trial result	Status
1	00:00:25.695	21%	Off track
2	00:00:18.355	21%	Off track
3	00:01:35.632	100%	Lap complete
4	00:01:31.521	100%	Lap complete
5	00:01:17.527	83%	Off track

表3 : Fumiaki Loop シミュレーション結果(2時間)

Trial	Time	Trial result	Status
1	00:00:45.419	60%	Off track
2	00:00:14.716	18%	Off track
3	00:01:17.434	100%	Lap complete
4	00:00:45.644	59%	Off track
5	00:01:17.042	100%	Lap complete

他のコースで走行させた結果,1時間学習させたモデルと2時間学習させたモデルは,どちらもほぼ全てのコースを完走することができた.

2つのモデルの性能差は,SOLA Speedwayのシミュレーション結果に大きく表れた.1時間学習させたモデルは,SOLA Speedwayのコースだけ,5回の試走行のうち1度も完走することが出来なかった.2時間学習させたモデルは,どのコースのシミュレーション結果も,1時間学習させたモデルよりも評価が上がり,SOLA Speedwayのコースでも,表4と表5を比較すると,モデルの性能が大きく向上したことが確認できる.これにより,学習時間を増やした方がより良い結果が得られることが分かった.

表4 : SOLA Speedway シミュレーション結果(1時間)

Trial	Time	Trial result	Status
1	00:00:11.205	12%	Off track
2	00:00:12.512	12%	Off track
3	00:00:10.932	13%	Off track
4	00:00:56.756	69%	Off track
5	00:00:11.706	12%	Off track

表5 : SOLA Speedway シミュレーション結果(2時間)

Trial	Time	Trial result	Status
1	00:00:57.069	100%	Lap complete
2	00:00:49.292	86%	Off track
3	00:00:54.037	100%	Lap complete
4	00:00:55.934	100%	Lap complete
5	00:00:53.636	100%	Lap complete

完走時間短縮を目指した場合,ステアリングをなるべく切らずにまっすぐ走る設定やカーブでのステアリング制御が鍵となると予測される.

中心線の進行方向に沿って自身も角度を変えるという報酬関数の設定に,カーブが少ないコースでステアリングの角度を小さくし,減速にペナルティを与える設定を加えると,他のコースでも完走率が安定し,よりタイムを縮められると思われる.

4.おわりに

本研究は,どの走行コースでも走行可能なモデルを作成することを目的として,Deep racerを用いて報酬関数の設定を行い,1時間学習したモデルと2時間学習したモデルを作成した.その後,各コースのシミュレーション走行をし,2つのモデルの性能評価,比較を行った.

結果として,SOLA Speedwayのコースで大幅な性能の向上を確認し,他のコースでも走行時間の短縮を確認できた.これにより,学習時間を長くした方が,性能は向上すると結論づけられる.

時間の都合上,アクションやハイパーパラメータの細かい設定は出来なかったが,報酬関数の設定を変えるだけでも,十分な成果を得ることが出来た.

今回は,中心線の進行方向に沿って自身も角度を変えるという報酬関数の設定が不十分でコースアウトをする場面が見られたので,カーブ時のステアリング制御を改善することで,他のコースでもより完走率を安定させ,完走時間を短縮させたい.

今後は,完走時間に焦点を当て,モデルカーの最大速度を上げ,速度を落とした際のペナルティ設定やカーブで減速する機能を実装し,より高性能なモデルの実現を目指す.また,作成したモデルの評価実験を増やし,学習モデルについての考察を深めたい.

参考文献

- [1] Waymo LLC, "Waymo : Home", <https://waymo.com/> (2019-2020).
- [2] Amazon Web Services, "AWS DeepRacer 報酬関数の入力パラメータ", https://docs.aws.amazon.com/ja_jp/deepracer/latest/developerguide/deepracer-reward-function-input.html (2020).
- [3] Amazon Web Services, "クラウドならアマゾンウェブサービス【AWS公式】", <https://aws.amazon.com/jp/> (2020).
- [4] AWS公式 Work Shop "Lab 1: Get ready to race by building your own AWS DeepRacer RL model!", <https://github.com/aws-samples/aws-deepracer-workshops/blob/master/Workshops/2019-AWSSummits-AWSDeepRacerService/Lab1/Readme-Japanese.md> (2020).