計測自動制御学会東北支部第337回研究集会(2022.6.27)
資料番号337-4

訓練データ選別によるエージェント学習の性能評価

Performance Evaluation of Agent Learning by Training Data Selection

○山地龍生*，釜谷博行*，工藤憲昌*，原元司**
Ryusei Yamachi*, Hiroyuki Kamaya*, Norimasa Kudoh*, Motoshi Hara**
*八戸工業高等専門学校，**松江工業高等専門学校
*National Institute of Technology, Hachinohe College,
**National Institute of Technology, Matsue College

キーワード: 深層学習(Deep Learning), 変分オートエンコーダ(Variational AutoEncoder),
共分散行列適応戦略(Covariance Matrix Adaptation Evolution Strategy), 学習エージェント(Learning Agent)
連絡先: 〒039-1192 青森県八戸市田面木字上野平16-1 八戸工業高等専門学校　産業システム工学専攻
Tel.: 0178-27-7283，E-mail: kamaya-e@hachinohe-ct.ac.jp

1. はじめに

自動運転や自律制御ロボットなどの基盤技術として、近年、学習エージェントが注目されている。エージェントとは、環境内で様々な試行錯誤を繰り返し、行動を最適化していく学習者である。しかし、エージェントの訓練には大量の学習データが必要であり、データに画像を直接用いると、学習に時間がかかってしまうという問題がある。そこで変分オートエンコーダ(VAE:Variational AutoEncoder)を用いる方法が提案されている。VAEは、ディープラーニングによる生成モデルの1つで、訓練データを元にその特徴を捉えて、潜在変数に変換する。この潜在変数を用いることで画像の次元圧縮ができ、学習時間を短縮できる。

一方、訓練データ中に性能向上に寄与しないものが含まれると、学習に時間がかかるという問題がある。

そこで本研究では、予め訓練データを選別することで、無駄な訓練を省き、学習の効率化を図ることを目的とする。本発表では、OpenAI Gymの学習問題において、9点の色判別による訓練データ選別の手法を提案し、性能評価を行う。

2. 学習環境

OpenAI Gym という、強化学習アルゴリズムのプラットフォーム内の課題である Car Racing を用いて実験を行う。これは、エージェントがアクセル・ブレーキ・ハンドルを調節し、コースを上手に走れるようにする課題である。

図１に学習アーキテクチャの全体図を示す[1]。観測データは、各時間ステップ t で変分オートエンコーダ(VAE)によって状態が圧縮され、潜在変数 z_t が生成される。コントローラへの入力は、この潜在変数 z_t であり、各時間ステップで混合密度ネットワーク(MDN-RNN：Mixture Density Network -Recurrent Neural Network)の隠れ状態 h_t と連結される。MDN-RNN は、現在の z_t と行動 a_t を入力として受け取り、自身の隠れ状態を更新して時刻 t+1 で使用される h_{t+1} を生成する。このように VAE、MDN-RNN、コントローラが相互作用すること

で、エージェントがコースを上手に走れるように学習を行う。

学習目的は、道路から外れずに制限時間内にできるだけ長い距離を走行することである。観測データ(状態)には、図 1 の左に示すような赤色のエージェントの真上から見た RGB 画像を用いる。灰色の部分は環状の道路である。行動は、エージェントがアクセル・ブレーキ・ハンドルを調節することである。道路は区切られていて、1 区画通過する毎に報酬が与えられる。報酬は、時間ステップ毎に-0.1、道路の 1 区画を通過する毎に 3.67 が与えられ、すべての区画を通過すると合計で 1,000 となる。評価には、1,000 ステップの行動で得られた報酬を用いる。

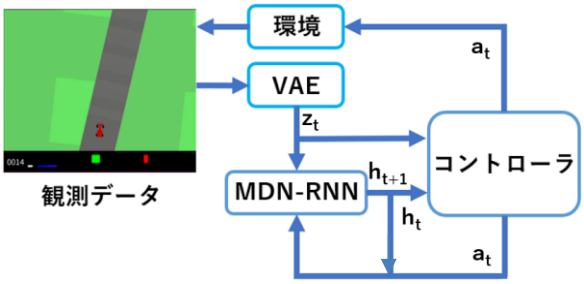

図 1　学習アーキテクチャの全体図

3. 学習の流れ

以下の 5 つのステップで学習を進める[2]。

Ⅰ. ランダムなロールアウトデータの収集

ここでは、エージェントが与えられたタスクとは無関係にランダムに環境を探索する。複数のエピソードをシミュレートし、時間ステップ毎の観測された状態、行動(ランダム)、報酬(-0.1 か 3.67)を格納する。ここでの狙いは、エージェントの行動によって、環境がどのように変化するのかに関するデータセットを構築することである。

図 2 は、あるエピソードの 40 番目のフレームを表している。画像は状態、a は[ハンドル, アクセル, ブレーキ]、r は報酬を表している。この図の場合、エージェントがハンドルを変えず、動かなかったため、報酬が-0.1 与えられたということを示している。

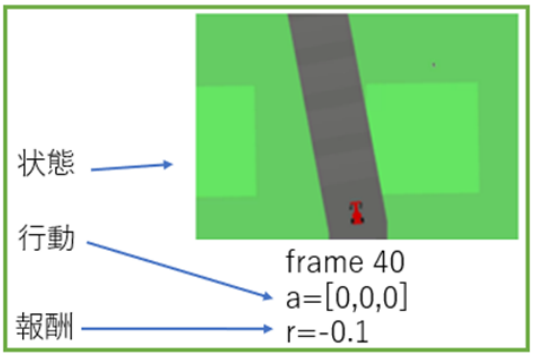

図 2　あるエピソードの 40 番目のフレーム

Ⅱ. VAE(Variational Autoencoder)の訓練

VAE とは、ディープラーニングによる生成モデルの 1 つで、訓練データを元にその特徴を捉えて、訓練データセットに似たデータを生成することができるモデルのことである。

ここでは、ランダムに集めたデータを使って、VAE を観測画像で訓練する。Ⅰで集めたデータを用いて学習することで、VAE はその状態を効率的に潜在変数としてとらえられるようになる。VAE の目的は、1 枚 64×64×3 (縦×横×RGB) の画像を正規分布するランダムな潜在変数(mu,log_var)に低次元化することである。mu(平均)と log_var(分散)で 2 次元の正規分布を表す。

図 3 は VAE の概略図である。入力画像と出力画像が同じになるように VAE を訓練する。VAE に 64×64×3 の画像が入力されるとエンコーダを介して、32 次元の潜在変数になる。そして、32 次元の潜在変数はデコーダを介して 64×64×3 の出力画像になる。図 4 にエンコーダのネットワーク構成を示す。

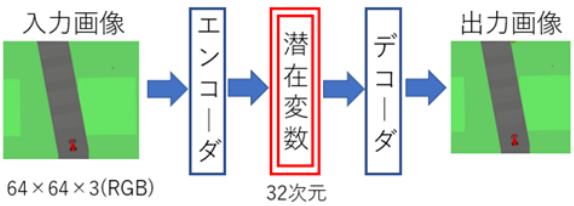

図 3　VAE の概略図

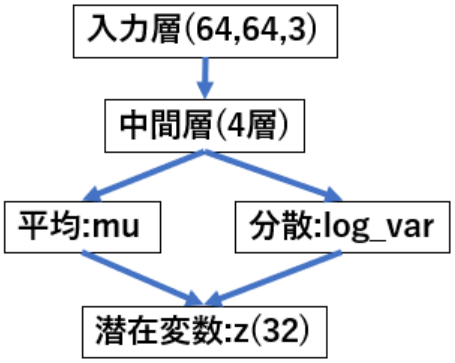

図 4 エンコーダのネットワーク構成

Ⅲ. MDN-RNN(Mixture Density Network - Recurrent Neural Network)訓練用のデータ収集

MDN は従来のニューラルネットワークを混合密度モデルと結合して得られたモデルのクラスのことである。MDN では、コスト関数に多峰性の混合ガウス分布でモデルされたものを用いる。こうすることで、確率的な事象も扱えるようになる。RNN は、時系列データを取り扱えるニューラルネットワークのことである。RNN は入力層・中間層・出力層から成り立っており、中間層の演算結果を出力するとともに、同じ演算結果を再び中間層に入力し再演算する。このことにより、以前に計算された情報を記憶しておくことができる。

ステップⅡで VAE が訓練されるので、これを用いて RNN 用の訓練データを生成することができる。このステップでは、ランダムなロールアウトデータすべてを VAE に渡し、各観測に対応する mu(平均)と log_var(分散)ベクトルを格納する。このエンコードされたデータは、すでに収集された行動と報酬とともに、MDN-RNN を訓練するのに使われる。

Ⅳ. MDN-RNN の訓練

MDN-RNN の目的は、1 ステップ後の未来を予測することであり、VAE が生成することが期待される将来の潜在変数の予測モデルとして機能することである。

このステップでは、各エピソードに対応する時間ステップ毎の mu、log_var、行動、報酬変数を読み込む。次に、MDN-RNN が z ベクトル、行動、報酬の入力から、次の z ベクトルと報酬を予測できるように訓練される。

図 5 の左は MDN-RNN のモデルである。この図のように現在の情報から未来を予測できるように訓練する。図 5 の右は RNN のネットワーク構成である。

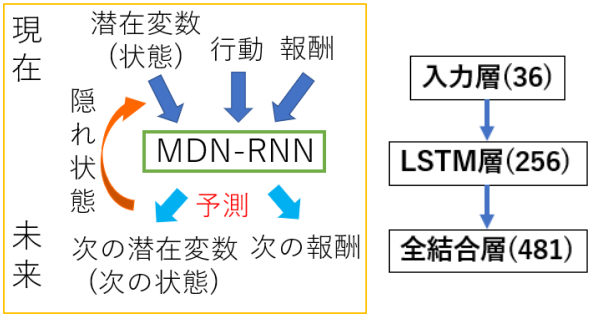

図 5 MDN-RNN モデル

Ⅴ. コントローラの訓練

ある状態でどのような行動をとれば、より良い報酬を得られるかを学習させることをコントローラの訓練の目的とする。

ここでは、訓練済みの VAE と RNN を用いてコントローラを訓練し、与えられた現在の z と RNN の隠れ状態 h から行動を出力できるようになる。

進化アルゴリズムの共分散行列適応進化戦略(Covariance Matrix Adaptation Evolution Strategy)を用いて訓練を行う。このアルゴリズムは、そのタスクに対して全体的なスコアが高くなる行動を生成するエージェントに報酬を与える。そうすることで、この望ましい振る舞いを未来の世代も継承する可能性が高くなり、スコアの向上が期待できる。

図 6 は、カーブを曲がる際の行動イメージである。アルゴリズム内のコントローラのパラメータが最適化されていくことで、エージェントを動かすコントローラは徐々に、青矢

印のような行動をとらずに、赤矢印の行動をとるようになりカーブを上手に曲がれるようになる。

進化アルゴリズムの処理ステップを以下に示す。

①エージェントの集団を作成し、各エージェントに対するパラメータをランダムに初期化

②以下の a~d を繰り返す

a. 環境内で各エージェントを評価し、複数のエピソードの平均を求める

b. 集団から最も良いスコアを持つエージェントを交叉させ、新しいメンバーを作る

c. 新しいメンバーのパラメータにランダム性を追加する

d. 新たなエージェントを追加し、パフォーマンスの悪いエージェントを取り除く

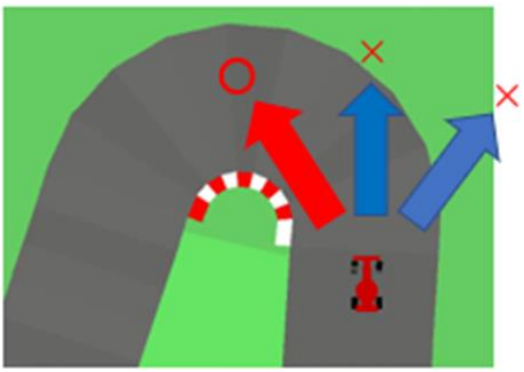

図 6　カーブを曲がる際の行動イメージ

4. 提案手法

収集したロールアウトデータを用いて、以下の 2 つのパターンでそれぞれ訓練を行う。

① 収集したすべての観測データを用いる場合

② エージェントが道路から完全に外れた観測データを削除する場合

エージェントが道路から完全に外れると、エージェントは正の報酬をもらえず、性能向上が期待できない。また、エージェントが道路から大きく外れると戻れることは少ない。そこで、②のようにエージェントが道路から完全に外れた場合には、以降のデータを削除することにした。こうすることで、訓練の効率化を図れると考えた。

観測データの選別は、エージェント周辺の 9 点の色を判別することで行う。9 点は図 7 に示す通り、エージェントを中心として、中心・上・右上・右・右下・下・左下・左・左上である。もし上記の 9 点がすべて緑であれば、完全に道路から外れたと認識する。一方、1 点以上灰色があれば、まだ道路の近辺にいると認識する。

実験では、上記の 2 つのパターンで VAE をそれぞれ訓練し、その後、コントローラを学習する。

図 7　色判別の箇所

5. 実験

5.1 実験で使用したパラメータ

学習の流れで示したⅡ,Ⅳ,Ⅴで使用したパラメータを以下に示す。

Ⅱ. VAE 訓練に使用するエピソード数:1000
　訓練エポック数:20

Ⅳ. 各訓練の繰り返しで、MDN-RNN に渡す
　エピソード数:3000
　訓練の繰り返し総数:1000

Ⅴ. 解を並列にテストするワーカー数：4
　世代ごとにテストするために各ワーカーに割り当てられる解の数:4
　平均報酬を計算するために、それぞれの解

がテストされるエピソード数:4
各エピソードの最大時間ステップ数:1000
その時点で最も良いパラメータセットの評価間の世代数:10

5.2 実験結果

4 で示した提案手法の通りに、2 つのパターンで実験を行った。

どちらの実験も、1 世代の訓練には約 12 分かかった。計算機のスペックは、プロセッサ: AMD Ryzen 5 3600xt (3.8GHz)、メモリ: 64GB、OS : Ubunts 20.04 である。

図 8 は、改良前の世代数とその世代での平均報酬を表している。学習初期には、報酬が負であるため、エージェントが道路外で動いている、もしくは動いていない。学習中期には、平均報酬が急激に増加している。学習終期には報酬が 800 程度になり、道路を一周できるようになった。

図 9 は、改良後の世代数とその世代での平均報酬を表している。報酬が全体を通して低く、最大で 230 程度である。また、図 8 の約 3 倍の世代の学習を行ったが、報酬の値はかなり低い。そして、平均報酬がこまめに増減を繰り返していることから、適切に行動を学習できていないことが分かる。なお、色判別プログラムの導入によって、②の訓練データ数は 3 分の 1 程度に減少した。

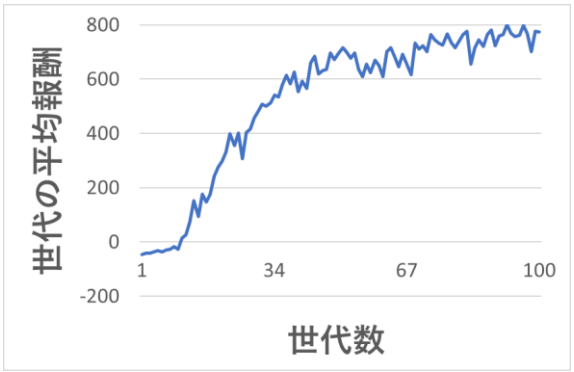

図 8 改良前の世代数と世代の平均報酬

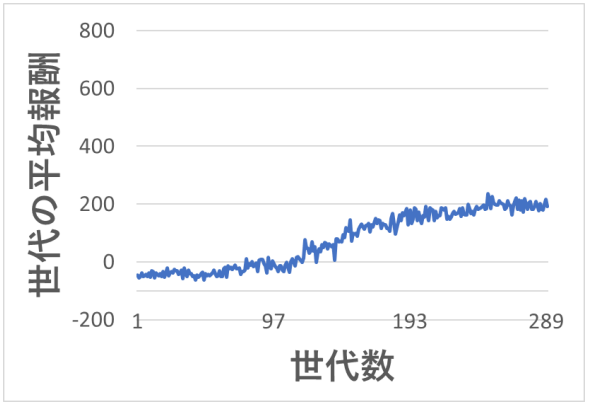

図 9 改良後の世代数と世代の平均報酬

6. まとめ

道路から外れた状況を除外するプログラムを作成し、観測データの選別はできた。しかし、改良後のエージェントが改良前よりも、性能が悪いという結果が出てしまった。これらの原因として、データ削除によるデータ不足（道路外の訓練データも性能向上に寄与している可能性がある）、プログラムの不具合などが考えられる。

現在のアルゴリズムでは、カーブの内側を走行した際に、数ステップ先に道路に戻れるデータも削除している。そこで、色判別の箇所について再検討が必要である。

参考文献

1) David Ha, Jürgen Schmidhuber : ” World Models”, https://arxiv.org/abs/1803.10122 (2018)
2) David Foster 著、松田晃一、小沼千絵訳 :” 生成 Deep Learning”, オーム社 (2020)