多様な状態予測によるモデルベース強化学習の改善

Improving Model-based Reinforcement Learning through the Prediction of Diverse State Transitions

堀内 優太 † 白浜 公章 ‡
Yuta Horiuchi Kimiaki Shirahama

1. 序 論

強化学習はモデルベース強化学習とモデルフリー強化学習の 2 つに分類することができる．それぞれの大きな違いはモデルが存在するか否かであり，モデルベース強化学習は環境の情報をモデルとして扱い，モデルと学習を行うことができるためサンプル効率が良いと注目されている．モデルフリー強化学習はモデルがないのでサンプル効率は良くなく，多くの学習データが必要だが実装及び調整が容易にできることが特徴である．モデルベース強化学習は注目されつつあるも，モデルを学習する際のデータの質や傾向に依存度が高いのが問題点とされており，モデルフリー強化学習と比較した場合モデルフリー強化学習の方がスコアが高いという報告が多い．本研究では近年発表されたモデルベース強化学習「DreamerV2」[2] に注目した．この DreamerV2 は 1 つのフレームに着目し，そこから将来を想像して行動を学習している．その結果一部のモデルフリー強化学習のスコアを上回り注目を集めた強化学習方法である．しかしながら 1 フレームの画像から 1 通りの将来しか想像しないため非効率だと考えた．この問題を解決するために，本研究では 1 つしか想像しなかった DreamerV2 を改良してノイズを付与し，他の状態を作成し，そこからさらに先を想像する手法を提案する．

2. 提案手法

DreamerV2 は 3 つのステップ，モデルの学習，行動の学習，データの収集によって学習を行っている．モデルの学習では画像を潜在変数で表し，潜在変数から画像に戻した時近い画像や状態が生成できるようになることが目的である．行動の学習では 1 つのフレームの画像を元に潜在変数に変換する．変換した潜在変数を元にどの行動をすれば報酬を得られるかを考えて行動を学習する．更新したモデルを使って実際の環境に影響を与え，データを収集して最初のステップに戻る．

2.1 ステップ 1: モデルの学習

以下ではモデルの学習について説明する．今, コンピュータゲームにおいて，各フレームでのプレイヤーの情報，敵の情報，舞台の情報などをまとめて状態と呼ぶ．そして，状態を要約した潜在変数を式 1 の h_t とし，再現モデルと呼ぶ．現在時刻 t において h_t は前時刻の h_{t-1},z_{t-1},a_{t-1} の 3 つのパラメータを元に決まる．初期値はランダムに決める．式 2 は表現モデルと呼び，z_t を予測するために使用する．ここで，z_t は，h_t，画像データ x_t を考慮しているのでより具体化した潜在変数である．式 4 から式 6 はそれぞれ画像 $\hat{x}_t$，報酬 $\hat{r}_t$，割引率 $\hat{\gamma}_t$ を h_t と z_t を元に予測する．つまり h_t と z_t を正確に予測できるとゲームの情報を 2 つの変数だけで予測することができる．

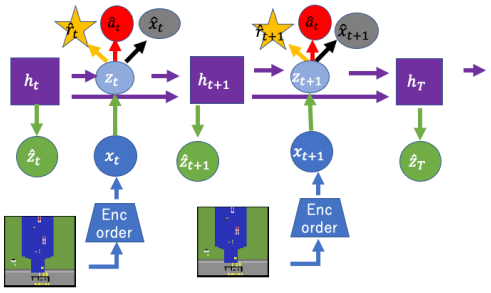

図 1: 潜在変数とモデルの予測の関係図

$$h_t = f_\phi(h_{t-1}, z_{t-1}, a_{t-1}) \quad (1)$$

$$z_t \sim q_\phi(z_t|h_t, x_t) \quad (2)$$

$$\hat{z}_t \sim p_\phi(\hat{z}_t|h_t) \quad (3)$$

$$\hat{x}_t \sim p_\phi(\hat{x}_t|h_t, z_t) \quad (4)$$

$$\hat{r}_t \sim p_\phi(\hat{r}_t|h_t, z_t) \quad (5)$$

$$\hat{\gamma}_t \sim p_\phi(\hat{\gamma}_t|h_t, z_t) \quad (6)$$

それぞれのモデルにはパラメータ ϕ の一部が使用されている．それぞれどの情報と組み合わせて報酬などを予測しているかの関係図のイメージ図を図 1 に載せる．z_t と $\hat{z}_t$ を使い, 想像モデル，報酬モデル，割引モデル，KL Ballancing これらを使い，損失関数を求める．損失が小さくなるような ϕ を探し，更新，これを時刻 T まで行う. 損失関数は式 7 に示す.

† 近畿大学, Kindai University
‡ 同志社大学, Doshisha University

$$L_\phi = E_{q_\phi(z_{1:T}|a_{1:T},x_{1:T})}\Bigg[\sum_{t=1}^{T} -lnp_\phi(x_t|h_t,z_t) \\ lnp_\phi(r_t|h_t,z_t) - lnp_\phi(\gamma_t|h_t,z_t) \\ +\beta KL[q_\phi(z_t|h_t,x_t)||p_\phi(z_t|h_t)]\Bigg] \tag{7}$$

学習はパラメータ ϕ の更新によって行われる．その ϕ を求めるために損失関数というものを使用する．パラメータ ϕ の損失関数は式 7 のようになる．また式中の T は集めたデータの最終時刻である．この式は前から，イメージモデルの損失，報酬モデルの損失，割引モデルの損失，KL 損失を表しており，学習の目的はこの期待値がもっとも低くなるような ϕ を探し，そのパラメータを使いモデルを更新する．それぞれの損失はそのベクトルになる確率のエントロピーを取っており，確率が低ければ低いほど値が大きくなる，KL の損失は引数 2 つの確率分布がどのぐらい似ているかの尺度のことであり，この値が大きいと 2 つの確率分布は似ていないということになる．

2.2 ステップ 2: ActorCritic 法による行動の学習

ステップ 2 は ActorCritic 法を使った行動の学習がメインだ．ActorCrtic 法は状態から行動を決める Actor とこれからどれくらいの報酬が得られるかを考慮する Critic，この 2 つがそれぞれ学習し，干渉しながらエージェントの行動を決める学習である．本研究では，モデルを使用して 1 フレームから状態価値や行動などの将来の状態を予測を行い，それらの情報を頼りに Actor，Critic の学習を行なっていく．それぞれ Actor はもっとも Critic が予測した報酬が大きくなるような行動を行い，Critic は Actor が行った行動によってどのような報酬が得られるかを予測する．予測にはそれぞれモデルを使用し，モデルのパラメーター ψ と ξ を使用する．

ステップ 2 は ActorCritic 法によって, パラメーター ξ と ψ の更新を行い，行動と状態価値の学習を行う．DreamerV2 がどのように 1 フレームの画像から将来を想像し，学習に必要なデータを予測しているかのイメージ図を図 2 に載せる．また式を式 8,9 に示す．

$$Actor : \hat{a}_t \sim p_\psi(\hat{a}_t|\hat{z}_t) \tag{8}$$

$$Critic : v_\xi(\hat{z}_t) \approx E_{p_\phi,p_\psi}[\sum_{\tau\geq t} \hat{\gamma}^{\tau-t}\hat{r}_\tau] \tag{9}$$

Critic 側，パラメーター ξ 側の学習方法は ξ をパラメーターにした MDP によって $\hat{z}_t$ の状態価値を求める．そして実際にモデルによって計算された報酬和との二乗誤差の期待値を求める．この期待値が低くなるような ξ を求めてパラメーターを更新する．

それぞれ Actor はもっとも critic が予測した報酬が大きくなるような行動を行い，Critic は Actor が行った行動によってどのような報酬が得られるかを予測する．それぞれモデルのパラメーター ψ と ξ を使用する．
次にパラメーター ψ と ξ の更新はステップ 1 と同様損失関数を使用する．

まず，Critic の損失関数に使用される時刻 t の時の状態価値の定義について説明する．式 10 がその式になる．

$$V_t^\lambda \approx \quad \hat{r}_t + \hat{\gamma}_t \begin{cases} (1-\lambda)v_\xi(\hat{z}_{t+1}) + \lambda V_{t+1}^\lambda & if \quad t < H \\ v_\xi(\hat{z}_H) & if \quad t = H \end{cases} \tag{10}$$

次にモデルのパラメータ ξ の更新を行うための損失関数について説明する．式は式 11 になる．

$$L(\xi) \approx E_{p_\psi,p_\phi}[\sum_{t=1}^{H-1}(v_\xi(\hat{z}_t) - V_t^\lambda)^2] \tag{11}$$

この式はモデルが考えた時刻 $\hat{z}_t$ においての状態価値と報酬を元に考えた状態価値の二乗誤差を求める．より良いモデルパラメータ ξ を求めるため，損失関数を使用し，その期待値が最も小さくなるような ξ を探す．

Actor 側は Critic 側のパラメータを使用し，行動モデルを使った期待値を計算する．その期待値が一番小さくなるような ψ を探す．Actor についても損失関数を使用する．使用されるパラメータは ψ を使用する．式は式 12 を使用する．

$$L(\psi) \approx E_{p_\phi,p_\psi}\Bigg[\sum_{t=1}^{H-1}(-\rho lnp_\psi(\hat{a}_t|\hat{z}_t)(V_t^\lambda - v_\xi(\hat{z}_t)) \\ -(1-\rho)V_t^\lambda - \eta H[a_t|\hat{z}_t])\Bigg] \tag{12}$$

実際に行われた行動のエントロピーを最適化し式に加えることによって探索を促す作用がある．損失関数は一番く小さくなるようにパラメータ ξ を決める．損失が 0 より小さい時，これは報酬を元に考えた状態価値とモデルが考えた状態価値の差がマイナスであることを示しており，モデルに差があるため良いモデルとは言えていない．次に 0 に近しい時は状態価値とモデルの価値はほぼ等しい状況にあるが同じ行動をしているとより良い行動を見つけれないことがあるため探索を行う必要がある．ここで使用されるのが正規化エントロピーである．これにより実際に行った行動が確率の低い，つまりエントロ

ピーが大きいものを加えることによって想像の中での行動で探索を促すようにしている Actor は Critic の想像する報酬を最大化することが目的であり，Critic は Actor が取った行動の報酬と想像する報酬ができるだけ小さくなるようすることが目的である．

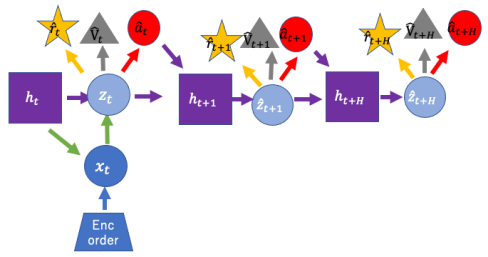

図 2: 将来の想像イメージ図

2.3 ステップ 3: データの取得

ステップ 3 では学習されたモデルを使用して，実際にゲームをプレイし各時点で行動を選択し，報酬や画像データを受け取る．それらをデータセットとしてまとめステップ 1 へ戻る．以上が DreamerV2 についての大まかなアルゴリズムである．

2.4 提案手法

提案手法ではステップ 2 とステップ 3 の間にノイズを加えてさらに学習を行うステップを加える．これは IDM(Imagination with Derived Memory)[3] の想像の拡張のアイデアを参考にする．DreamerV2 で問題点は 2 つあると考える．1 つ目は将来考える状態が 1 つだけである点，もう 1 つは新しい状態を考えない点である．本研究では後者新しい状態を考える点に重点を置く．また人間はイメージトレーニングのように実際に経験した状態を元に似たような全く新しい状態で自分がどのように行動するか事前に学習することができる．新しい状態を生成する案として状態を定義している潜在変数 z_t の値を変化させることを考え，IDM を参考にする．またノイズの生成方法として学習データの潜在変数の値のガウス分布を取り，ノイズ δ をサンプリングする．そして，定数 c を乗じた $c\delta$ をその値を潜在変数 z_t に加えた．これによって行動を学習する時に潜在変数にノイズを加えた別の似たような状態の環境でさらに学習ができると考えた．イメージ図を図 3 に示す．

3. 結果・考察

3.1 研究環境

使用したゲームは OpenAI Gym から提供されているレトロゲーム環境「RiverRaid」を使用しており，どれ

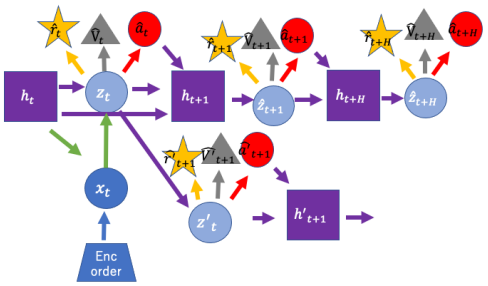

図 3: 提案手法

くらい前に進めたか，またどれくらい障害物を壊したかで報酬が決まる．また報酬などはゲームによって様々である．ステップ数は DreamerV2 を参考に 8000 万ステップを目標値として実行した．ハイパーパラメータは DreamerV2 と同様のものを使い，ノイズを増減するハイパーパラメータ c は 1 から 3 倍で設定を行なった．

3.2 結果・考察

エージェントがどれくらいが報酬を受け取れたかを図 4 に示す．緑色が提案手法であり，オレンジ色が DreamerV2 である．縦軸がどれくらい 1 ゲームで報酬を受け取ったかを表しており，横軸はどれくらいのステップ学習したかになっている．結果として提案手法は DreamerV2 のスコアを少しだけ上回る結果となった．また World-Model を構成しているモデルの損失の合計値，Critic が想像した値，Actor の損失関数の値を図 5，図 6，図 7 に示す．それぞれを見比べる．図 5 は下がれば下がるほど画像を正確に潜在変数に変換できていることになり，初期段階は DreamerV2 がよかったが後半になるほど提案手法が良い結果となった．図 6 は上がれば上がるほど将来の報酬が良いと想像できている．この値が大きいと Actor 側に反映された後データ収集でゲームのプレイ時間が長くなりデータが増えるため良い影響を与えられる．こちらも後半になるにつれて提案手法が良い結果となっている．図 7 では Actor の損失を示しており，この値の大小により良いか悪いかが決まるわけではない．式 12 のように取った行動の情報量を考慮して新しい行動を促すこともしているため，大きな値があると新しい行動を取ろうとしており，小さければ小さいほど報酬の価値を重点的に見ていたということになる．

4. 結論・今後の展望

結論としては提案手法は DreamerV2 を少し上回り良い結果が得られたと考える．今回はノイズを加えて新しい状態を考えることが 1 回だけだったが複数回するとどのようになるか検討する必要があり，IDM のようにそれらの学習をどれくらい学習に含めるかを調整する

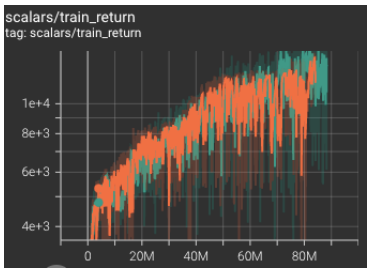

図 4: 提案手法と DreamerV2 の受け取った報酬

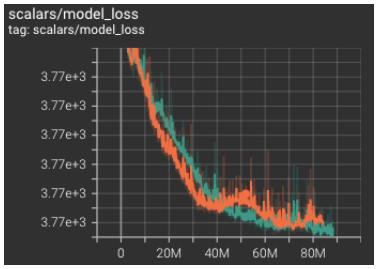

図 5: 提案手法と DreamerV2 の WorldModel の損失値

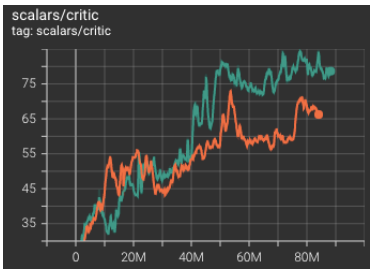

図 6: 提案手法と DreamerV2 の Critic が想像した価値

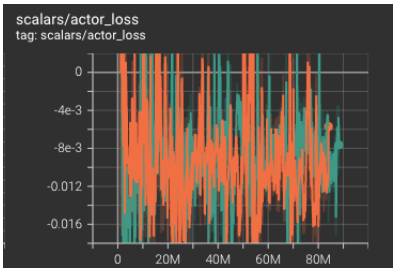

図 7: 提案手法と DreamerV2 の Actor の損失値

Evaluater の実装も行いたいと考えている．これによりより学習が早く進んだモデルベース強化学習の提案をしたいと考えている．

参考文献

1) Richard S. Sutton and Andrew G. Barto :Reinforcement Learning, A Bradford Book,2018

2) Danijar Hafner, Timothy Lillicrap. Jimmy Ba, Mohammad Norouzi: Dream to Control: Learning Behaviors by Latent Imagination Proc. of ICLR 2020, 2020

3) Yao Mu, Yuzheng Zhuang, Bin Wang, Guangxiang Zhu, Wulong Liu, Jianyu Chen, Ping Luo, Shengbo Eben Li, Chongjie Zhang and Jianye Hao: Model-Based Reinforcement Learning via Imagination with Derived Memory, Proc. of NeurIPS 2021, 2021