

# ゲーム対戦戦略をプレイヤー習熟度へ適応させる機械学習機構の設計

A Machine Learning Facility for Adapting Competitive Game Strategies to Players' Proficiency

竹内 大輔\* 野呂 昌満† 沢田 篤史‡

あらまし ゲーム AI の設計において、プレイヤーモデリングは重要な課題である。中でもプレイヤーの習熟度に応じて対戦戦略を柔軟に適応させる仕組みは、魅力的なゲームを構築するための鍵である。本研究の目的は、プレイヤーの時間的習熟を予測し、それをゲームエンジンにおける対戦戦略の柔軟な変更に応用できる共通基盤を構築することである。この目的を達成するために、LSTM (Long Short-Term Memory) に基づく習熟度学習器を設計し、この学習器を組み込んだソフトウェアアーキテクチャを提案する。簡単なターン制 RPG を対象とした実験により、提案する機械学習器の有効性と妥当性を確認した。

## 1 はじめに

### 1.1 研究背景と目的

ゲーム AI の設計において、プレイヤーのモデリングは重要な課題であり、その中でプレイヤーの状況を考慮することの重要性が強調されている [1]。プレイヤーはゲームをプレイするにつれて、戦略を覚え、時間経過とともに習熟し、上達する。このような時間的習熟はプレイヤーモデリングの対象の 1 つである。これを考慮することでゲームを魅力的にできる可能性があるが、これまであまり行われてこなかった。

本研究の目的は、プレイヤーの時間的習熟を予測して対戦戦略を切り替える仕組みを提案することである。すなわち、過去のプレイヤーの行動選択履歴から習熟を予測する機械学習器を設計し、予測結果を利用して動的にゲーム戦略を切り替えるためのアーキテクチャを設計する。本研究で設計する機械学習器およびそれを組み込んだアーキテクチャは、一般的なゲームエンジンの構成要素 [2] [3] のうち、入力と AI を代替するものである。

### 1.2 ターン制 RPG

本研究では、ターン制 RPG を題材にし、時間的習熟に基づく対戦戦略の切り替え機構を実現する。ターン制 RPG は、戦闘の場面において、プレイヤーとプレイヤーに相対する CPU (敵 CPU) の行動タイミングが隔離されているという特徴を持つ。この特徴により、ゲームに対する入出力を順序立てて考えることができるので、時間的習熟を考慮したプレイヤーのモデリングが行いやすくなる。

ターン制 RPG では、行動選択の種類、行動選択時の戦況、敵 CPU の行動パターンが、プレイヤーの次の行動選択に影響する。戦況、すなわちプレイヤーと敵 CPU の体力であるヒットポイント (HP) の状況と敵 CPU が選択する行動に応じて、プレイヤーが適切な種類の行動を選択しているか否かを観測することで、習熟を予測できる。

本研究では RPG 戦闘のルールを次のように想定している。

- プレイヤーと敵 CPU の HP 初期値は 100 ポイント
- プレイヤーと敵 CPU は交互に行動を選択、それに応じて双方の HP を変化
- どちらかの HP が 0 になるまで行動選択を繰り返す

\*Daisuke Takeuchi, 南山大学

†Masami Noro, 南山大学

‡Atsushi Sawada, 南山大学

- 選択可能な行動の種類は3種類（攻撃、防御、回復）、攻撃強度は2種類（弱、強）ここで、敵CPUの攻撃に対する防御、防御に対する回復、回復に対する攻撃が有効な行動選択であり、この理解が習熟に影響する。

### 1.3 習熟度と時間的習熟の予測

本研究では、プレイヤーの習熟度を、選択した行動が敵CPUを不利にする程度であるとする。時間的習熟は時間経過に伴う習熟度の変化であり、これを予測することで、プレイヤーに適した難易度への戦略調整を支援する。

時間的習熟の予測にはLSTM（Long Short-Term Memory）を利用する。上述の通り時間的習熟は習熟度の変化であり、その予測には過去の学習内容を利用する必要がある。また、個々のプレイヤーに適応した柔軟な予測が必要であることから、LSTMの利用が適切であると判断した。

## 2 習熟度を考慮したプレイヤーモデリングとその技術課題

プレイヤーの行動特性とその変化を予測する研究はこれまでも多く行われてきた。Minらは、プレイヤーの目標認識の予測を目的としたプレイヤーモデリングを行っている [4]。Valls-Vargasらは、時間経過によるプレイヤーのプレイスタイルの変化を予測するプレイヤーモデリングを行っている [5]。北らは、習熟を用いて肢体不自由者の補助を行うシステムの開発を行っている [6]。

これら既存の技術も参考にし、時間的習熟の予測、および予測結果を用いた戦略変更を可能とする仕組みを持つゲームエンジンを実現するために、本研究では次の技術課題を設定する。

1. 習熟度予測機構を統合したゲームエンジンアーキテクチャの設計
2. 習熟度予測機構の詳細設計
3. 妥当性の確認

以下、3節では技術課題1,2について、4節では技術課題3について詳述する。

## 3 習熟度に基づく対戦戦略適応を行う学習機構の設計

### 3.1 習熟度予測機構を統合したゲームエンジンアーキテクチャの設計

図1に、一般的なゲームエンジンアーキテクチャ [7] における本研究の位置づけを示す。本研究で設計する習熟予測機構（以下、学習器と呼ぶ）は、プレイヤーから入力を得る機能と、予測した習熟に応じて戦略を変更する機能を持ち、Front EndとAIに該当する。図の右側には、学習器におけるデータのやり取りを示す。

学習の際の振る舞いを図2に示す。テストデータは、プレイデータのうち、プレイ時に想定される入力データと、それに対する最適な出力データをまとめたものである。テストデータに対する習熟予測に基づくAIの行動変更結果と、テストデータとの比較を行うことで学習する。

運用の際の振る舞いを図3に示す。プレイヤーの行動選択内容と、ゲーム内パラメータのうち習熟予測に必要なデータが学習器に入力され、習熟予測が行われる。その結果に応じ、AIが対戦戦略を変化させ、新たな戦略に基づく行動を出力する。

### 3.2 習熟度予測機構の詳細設計

#### 3.2.1 習熟予測に用いるデータの検討

時間的習熟を予測するために用いるデータは、表1に示す行動傾向と戦績である。直感的に、プレイヤーの習熟は、そのプレイ方法の変化と勝敗に現れると考えることができる。行動傾向は、プレイヤーがどういった行動を好んで選択したかを示すものであり、プレイ履歴から取得する。戦績は、プレイヤーと敵CPUとの間の勝敗の系列である。

これらのデータはいずれもターンや戦闘数を経るごとに追加され、ターンごと、

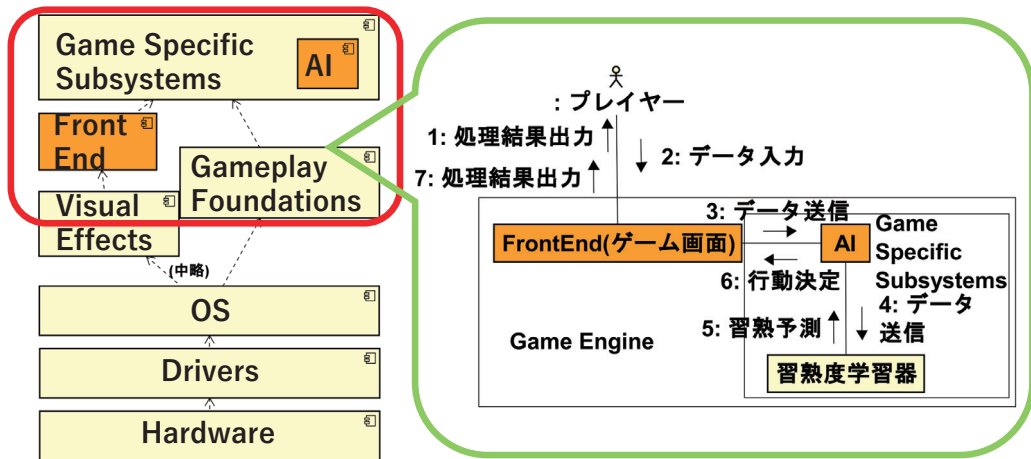


図1 ゲームエンジンアーキテクチャ [7] における提案手法の位置づけ

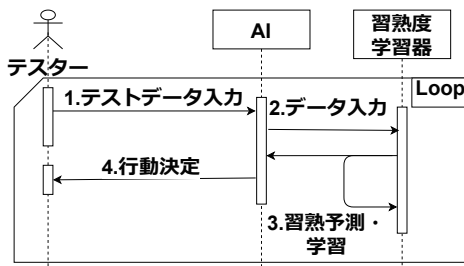


図2 コンポーネントの動的振る舞い(学習時)

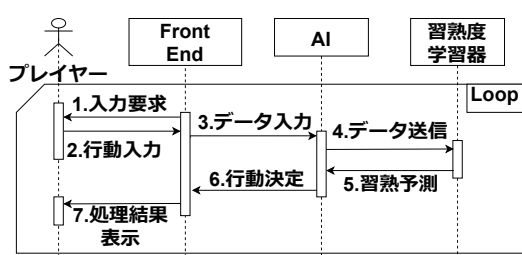


図3 コンポーネントの動的振る舞い(運用時)

表1 時間的習熟の予測に用いるデータ

データ種類	説明
行動傾向	プレイヤーの選択した行動の記録：行動の種類 (int), 強度 (int) の系列 (array)
戦績	プレイヤーの勝敗 (勝の場合 1・敗の場合 -1 : int) の系列 (array)

あるいは戦闘数ごとの連続した時系列データとして扱うことができる。これらは、プレイヤーが過去にどんな行動を選択し、その結果が好ましいものだったのか、そうでなかったのかを示すデータであり、学習器の入力として不可欠である。

### 3.2.2 LSTM の設計

学習器は、3.2.1 節で示した時系列データを学習し、プレイ時点におけるプレイヤーの習熟度の予測値を出力する。ゲームエンジンはこの予測値に基づいて、プレイヤーとの対戦戦略を変化させる。学習器の設計には、これまでの試行経験 [8] に基づき、LSTM を採用した。

図4に学習器の設計結果を示す。学習器内の各種パラメータは、実験において最も望ましい結果が得られたものを採用した。図中の変数  $t$  は、プレイ時間の経過に伴い増加するターン数を示す。学習器への入力、プレイヤーの選択行動、現在の習熟度、現在のパラメータ (HP) である。習熟度、HP は、ゲーム内で数値データとして管理しているものを用いる。これらの数値データをベクトル  $X(t)$  として入力

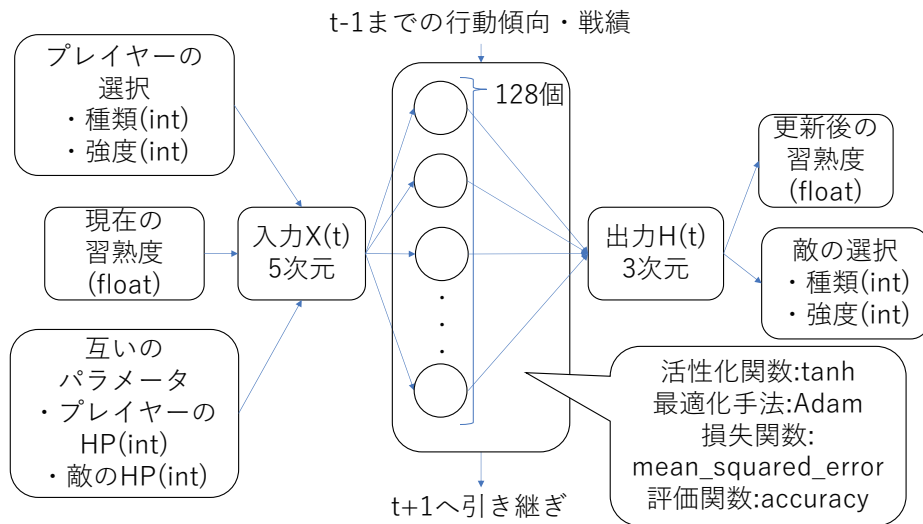


図4 習熟度学習器 (LSTM) の設計

する。時系列データを用いた予測において、プレイヤーの習熟度、次のターンの行動の予測を行い、敵CPUの行動を決定する。予測の結果、更新された習熟度、選択された敵の行動をベクトル  $H(t)$  として出力する。入力されたプレイヤーの行動、勝敗結果を、それぞれ行動傾向、戦績として、次のターンの予測に使用する。ただし、古すぎる行動や戦績は、不適切な予測を行ってしまう可能性があるため用いない。この設計により、入力データを時系列データに基づいて予測し、プレイヤーの習熟に応じた出力を得る。

## 4 実験による妥当性の確認

### 4.1 実験内容

前節で示した解決方法が本研究の目的に照らし妥当であるかを実験により確認する。妥当性の確認は、実際のゲームプレイを模した学習器の試験運用結果を分析することで行う。実験の手順は次の通りである。

- 3.1 節で述べたテストデータを用意し、学習器を学習させる。
- 学習済みの学習器を用いて、1人のプレイヤーとRPGの戦闘を行う。
- 2をプレイヤー1人につき50回行わせ、戦闘終了までにかかったターンと勝率の推移をグラフにする。
- 別のプレイヤーに、2を行わせる。
- 結果のグラフから学習の精度などを評価する。

ステップ1では学習器の学習(図2)を、ステップ2では学習器の運用(図3)を行う。個々のプレイヤーに学習器が適応するためには、十分な予測が必要であることから、ステップ3では1人のプレイヤーにつき50回戦闘を行わせた。

ステップ5で学習の経過をグラフに示し、グラフの形状で妥当性を評価する。実験では、プレイヤーの習熟度の変化を観測し、最終的な習熟度が高いプレイヤーと、全ての戦闘を通して低いままのプレイヤーに分類した。この実験では、習熟度を0から500の値域を持つ実数とした。戦闘中に習熟度400を超えたプレイヤーを習熟度の高いプレイヤー、それ以外を習熟度の低いプレイヤーとして分類した。

それぞれのグラフの形状を以下の仮説に基づいて評価する。

- 習熟度が高いプレイヤーの場合：勝敗が決まるまでに必要なターン数が増加、



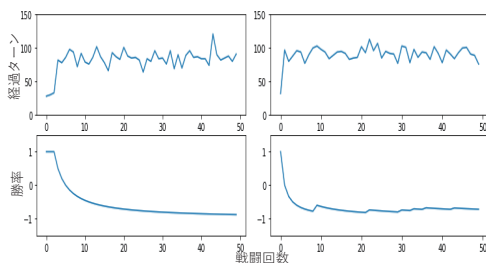


図5 経過ターンと勝率の推移 (習熟度が高いプレイヤー)

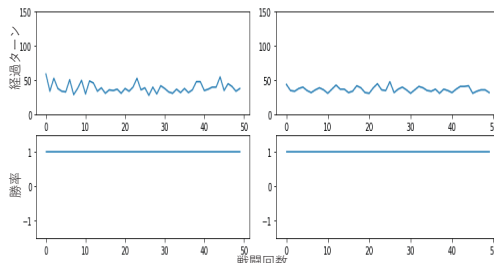


図6 経過ターンと勝率の推移 (習熟度が低いプレイヤー)

回数を重ねるごとに勝率が低下。

- 習熟度が低いプレイヤーの場合：勝敗が決まるまでに必要なターン数が減少，回数を重ねるごとに勝率が上昇。

これらと実験結果が一致していれば設計が妥当であると判断する。本研究では，異なる50人のプレイヤーによる実験を行い評価を行った。

## 4.2 実験結果

図5に習熟度の高いプレイヤー，図6に習熟度の上がらなかったプレイヤーに対する実験結果を抜粋したものを示す。それぞれ図の上部のグラフが，戦闘終了までにかかったターンの推移である。縦軸が経過ターン，横軸が戦闘回数を示している。下部のグラフが，プレイヤーの勝率の推移である。縦軸が勝率，横軸が戦闘回数を示している。

習熟度の高いプレイヤーの経過ターン数から，経過ターンが100ターン前後を維持していることが分かる。また，勝率は戦闘回数が増えるごとに振動し，0から-1の間に収束しており，プレイヤーが勝ちと負けを繰り返していることが分かる。これらから，学習器はプレイヤーの習熟度が上昇していることを予測し，これに伴って難易度を上昇させるために，対戦戦略を変更できていると言える。

一方で，習熟度の低いプレイヤーの経過ターン数から，50ターンを下回っていることが分かる。また，勝率からはプレイヤーが勝利を繰り返していることが分かる。これらから，学習器はプレイヤーの習熟度が上昇しないことを予測し，難易度を下げるために，対戦戦略を変更できていると言える。

## 5 考察

本研究では，ゲームプレイヤーの習熟度を予測するためにLSTMを用いて学習器を設計した。また，習熟予測に用いるデータとして，プレイヤーの行動選択の傾向と，戦績の2つのデータを採用した。これらを用いた学習と習熟度予測の結果，プレイヤーの習熟度に応じた戦略変更が可能であることを確認できたことから，この方針はおおむね妥当であったと考えている。

本研究で対象としたターン制RPGの習熟予測においては，自動翻訳に使用されるEncoder-Decoderモデル[10]と同様，直前のターンまでの可変長の時系列データを用いて予測を行うことが有効である。本研究では，このような判断からLSTMを選択し，実験によりおおむね良好な結果を得た。

一方，将棋をはじめとする対戦ゲームのモデリングには，CNNが使用される傾向にある[9]。これらCNNを用いた成果の習熟度予測への適用可能性については今後の課題としたい。

既存研究との関連について，Minらの研究[4]では，LSTMを用いた目標認識を行っているのに対し，本研究では同様のLSTMを用いてプレイヤーの行動を予測し

た. Valls-Vargas らの研究 [5] では, プレイヤーのプレイスタイルの変化のみを予測しているのに対し, 本研究では時間経過によるプレイヤーの習熟を考慮した.

## 6 おわりに

本研究の目的は, プレイヤーの習熟を予測して行動を切り替える仕組みの提案とその有効性の確認を行うことである. 目的の達成のために3つの技術課題を設定し, それぞれについて達成方法を提案した. 実験による評価では, これらの着想がおおむね妥当であることが示された.

一方で, 提案する学習器は, 入力ベクトルの次元数が少なく, 時系列データに重みを付与しないなど, 単純な設計である. より複雑なゲーム設定への対応, プレイヤーの習熟をより多様な側面から予測することを可能にするためには, 説明変数の追加と, アテンションの導入も検討課題である.

習熟度の予測に LSTM を利用するという着想は, 今後の研究でも引き続き利用できると考えているが, 入力データや再帰するデータを変更する場合には, 学習器を再度設計し直す必要がある. また, LSTM 以外のニューラルネットワークに対する枠組みと比較する必要もある.

習熟度学習器を組み込んだアーキテクチャに基づく, 実用的なゲームエンジンの構築も今後の課題である. ターン制 RPG 以外のゲームに対しても適用できるようにするためには, 本研究で検討外としたリアルタイム性や, プレイヤーの非同期的な行動についても考慮しなければならない. 多様なゲームに共通して使用できるような習熟度予測と利用の枠組みの構築も今後の研究目標としたい.

**謝辞** 本研究の一部は, JSPS 科研費 (基盤 (C) 20K11759), 2021 年度南山大学パッへ研究奨励金 I-A の助成による.

## 参考文献

- [ 1 ] D. Hooshyar, M. Yousefi, H. Lim: “Data-Driven Approaches to Game Player Modeling: A Systematic Literature Review”, *ACM Computing Surveys*, Vol. 50, No. 6, Article 90, 2018.
- [ 2 ] P. S. Paul, S. Goon, A. Bhattacharya: “History and Comparative Study of Modern Game Engines”, *International Journal of Advanced Computer and Mathematical Sciences*, Vol. 3, Issue 2, pp. 245-249, 2012.
- [ 3 ] M. Toftedahl, H. Engström: “A Taxonomy of Game Engines and the Tools that Drive the Industry”, *Proc. of the 2019 DiGRA International Conference: Game, Play and the Emerging Ludo-Mix*, 2019.
- [ 4 ] W. Min, B. Mott, J. Rowe, B. Liu, J. Lester: “Player Goal Recognition in Open-World Digital Games with Long Short-Term Memory Networks”, *Proc. of the 25th International Joint Conference on Artificial Intelligence (IJCAI-16)*, pp. 2590-2596, 2016.
- [ 5 ] J. Valls-Vargas, S. Ontanón, J. Zhu: “Exploring Player Trace Augmentation for Dynamic Play Style Prediction”, *Proc. of the 11th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pp. 93-99, 2015.
- [ 6 ] 北佳保里, 加藤龍, 横井浩史: “習熟度を考慮した自己組織的動作識別法の構築”, 日本ロボット学会誌, Vol. 28, No. 7, pp. 783-791, 2010.
- [ 7 ] J. Gregory: *Game Engine Architecture, Third Edition*, A K Peters/CRC Press, 2018.
- [ 8 ] 竹内大輔, 野呂昌満, 沢田篤史: “ゲームプレイヤーの習熟度に応じた対戦戦略の変更を可能とする機械学習器の設計”, 電子情報通信学会技術研究報告 (知能ソフトウェア工学), Vol. 121, No. 35, pp. 7-12, 2021.
- [ 9 ] 和田悠介, 五十嵐治一: “将棋の局所評価関数におけるディーブラーニングの応用”, 第 22 回ゲームプログラミングワークショップ論文集, pp. 244-249, 2017.
- [ 10 ] 林英里果, 竹本有紀, 石川由羽, 高田雅美, 城和貴: “近代文語体と現代口語体の自動翻訳への試み”, 情報処理学会研究報告 (数理モデルと問題解決), Vol. 2018-MPS-121, No. 18, pp. 1-6, 2018.