

RNNを用いたSteamGameの接続数の予測に関する研究

20A3058
指導教員

川又 翔
二宮 洋

1. はじめに

近年、ゲーム産業の発展より、家庭用ゲーム機やPCも高性能になり、様々なゲームが発売されている。プレイヤー数の増加に伴い、ゲームのサーバーがダウンするなどの問題も増加している。

そこで本研究では、ゲームの値段やプレイヤー数を入力とし、次月のプレイヤーの最大同時接続数の増加率を予測するAIを実装し、この問題の解決を試みる。提案手法では、Popularity of games on Steam[1]というデータセットと SteamDB[2]から取得したデータを、時系列データを扱うニューラルネットワークである Recurrent Neural Network(RNN)で学習する。

実験として、RNN と RNN を改善したモデルである Long-short Term Memory(LSTM)を用い、次月のプレイヤー数を予測する AI を実装しその結果を比較する。また、AI に入力する時系列の長さを変更し、その結果を比較する。

2. Recurrent Neural Network

RNN は、過去のニューラルネットワークの出力を再帰的にニューラルネットワークに入力する。これにより時系列データの性質を十分に学習することができる。RNN は、過去の情報を加算することで過去の情報を保持している。そのため、長期の過去の情報を保持することができない問題がある。LSTM は、過去の記憶情報をどれだけ忘れるかを判断するゲート、新たな記憶をどれだけ記憶するかを判断するゲート、記憶している情報をどれだけ出力するか判断するゲートを用いることで、RNN と比べ性能を改善している。

3. SteamGame の接続数増減に対する時系列予測

本研究では、ゲームの値段やプレイヤー数を入力とし、次月のプレイヤーの最大同時接続数の増加率を予測する AI の実装を試みる。提案手法では、Popularity of games on Steam(PGS)[1]というデータセットを用いる。PGS は、ゲーム販売プラットフォーム Steam[3]で販売されている 1,200 種類のゲームに対してプレイヤーの最大同時接続数を月ごとに記したデータセットである。また、SteamDB[2]から各月の Steam の累積プレイヤー数と、各月のゲームの平均価格を取得した。本研究では、 M_F は PGS においてゲームFの情報の蓄積を開始した月から経過した月の数、 P_{F,M_F} を M_F におけるゲームFのプレイヤーの最大同時接続数、 S_{M_T} を M_T における Steam の累積プレイヤー数、 C_{F,M_F} を M_F におけるゲームFの平均価格、 G_{F,M_F} を

M_F におけるプレイヤーの最大同時接続数の増加率とする。 G_{F,M_F} を(1)に示す。

$$G_{F,M_F} = \left(\frac{P_{M_F+1}}{P_{M_F}} \right) \times 100 - 100 \quad (1)$$

AI の学習に用いる入力 $\mathbf{X}_{F,n,k}$ を(2)、教師信号 $\mathbf{t}_{F,n,k}$ を(3)に示す。

$$\mathbf{X}_{F,n,k} = \begin{bmatrix} P_{F,n+1} & C_{F,n+1} & S_{F,n+1} \\ P_{F,n+2} & C_{F,n+2} & S_{F,n+2} \\ \vdots & \vdots & \vdots \\ P_{F,n+k} & C_{F,n+k} & S_{F,n+k} \end{bmatrix} \in \mathbb{R}^{k \times 3} \quad (2)$$

$$\mathbf{t}_{F,n,k} = [G_{F,n+1}, G_{F,n+2}, \dots, G_{F,n+k}]^T \in \mathbb{R}^k \quad (3)$$

ここで、 k は入力するデータの月の数である。学習に使用するモデルを図1に示す。

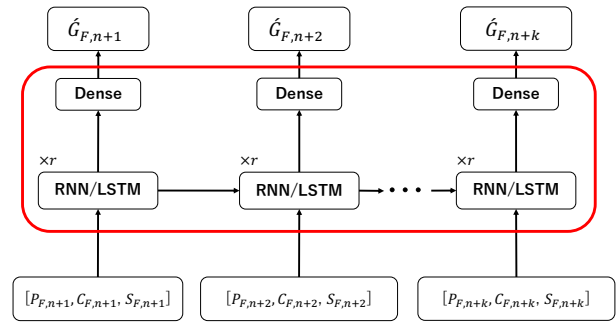


図1 学習に使用するモデルの構造

ここで、 r はモデルのRNN, LSTMの層の数である。提案手法のモデルでは、各月に対し次月のプレイヤーの最大同時接続数の増加率を予測する Many-to-Many 構造を採用した。

4. 実験

4.1 実験1

本実験では、RNN と LSTM を用い、次月のプレイヤーの最大同時接続数の増加率を予測するAIを学習し、その結果を比較する。本実験で使用するモデルのRNN, LSTMの層の数は1とし、ユニット数を1,024、活性化関数をReLU関数とした。学習に使用するゲームFはCSGOとした。CSGOのデータは84ヶ月分となっており、この内ランダムに選択した10ヶ月分のデータを検証データとした。また、損失関数を平均二乗誤差、最適化手法をAdam(学習率:0.0001)、学習反復回数を1,000とした。入力する時系列の長さを1から24とした学習結果を図2に示す。ここで、横軸は入力するデータの月の数、縦軸は検証データ

に対する誤差，破線が RNN，実線が LSTM である．図 2 から，RNN を使用したモデルの検証データに対する誤差が，LSTM を使用したモデルの検証データに対する誤差と比べ，小さな誤差を得ていることがわかる．これは LSTM の忘却ゲートの過去の情報を更新することが返って精度を低下させてしまう原因であると考えられる．次月の一ヶ月の予測を行う際に，過去の情報が忘れすぎて新しい情報に更新してしまい，予測精度が落ちてしまうと考えられる．この実験の結果から，次の実験では RNN を用いる．

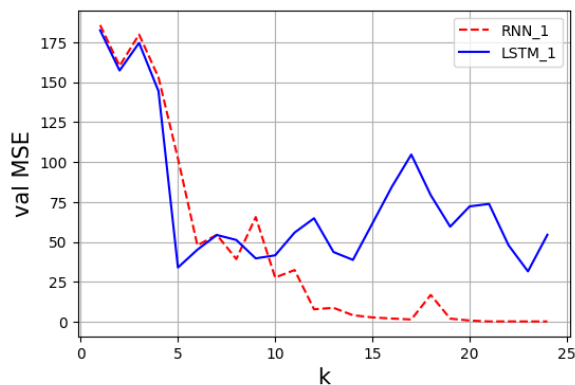


図 2 実験 1 の結果

4.2 実験 2

次に，モデルで使用する RNN を多層化し，その結果を比較する．一般に，ニューラルネットワークは中間層の数を増やすことで性能が向上することが知られている．そこで，本実験では中間層の RNN の層の数を 4 としたモデル構造と実験 1 の RNN のモデルを比較した．次月のプレイヤーの最大同時接続数の増加率を予測する AI を学習する．実験条件や学習に使用するデータは実験 1 と同様である．図 3 に学習結果を示す．ここで，横軸は入力するデータの月の数，縦軸は検証データに対する誤差，破線が中間層の RNN の層の数が 1，点線が RNN の層の数が 4 のモデルである．図 3 から，RNN の層の数が 1 のモデルは，入力するデータの月の数が 22 の場合に最も低い誤差を，RNN の層の数が 4 のモデルは，入力するデータの月の数が 23 の場合に最も低い誤差を得たことを確認した．また，モデルの RNN の層の数を増やすことで検証データに対する誤差が小さくなることを確認した．さらに，RNN の層の数が 4 のモデルは，入力するデータの月の数 k が 17 以上であれば，RNN の層の数が 1 のモデルの最も低い誤差より小さな誤差を得られていることを確認した．このことから，RNN を多層化することで，入力するデータの月の数を減らしても同等の性能が得られると考えられる．

最後に，RNN の層の数が 1，入力するデータの月の数が 22 のモデルと，RNN の層の数が 4，入力するデータの月の数が 17 のモデルの検証データに対する予測を図 4，図 5 に示す．ここで，縦軸はプレイヤーの最大同時接続数の増加率，横軸は予測する月，○が教師信号，×が予測である．

図 4，図 5 から，どちらも同等の予測結果が得られており，このことから RNN を多層化することで，入力するデータの月の数を減らせると考えられる．

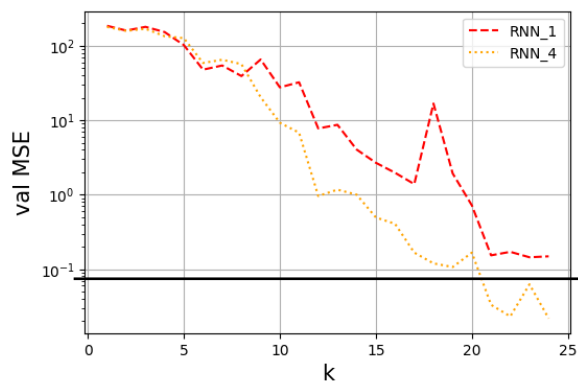


図 3 実験 2 の結果

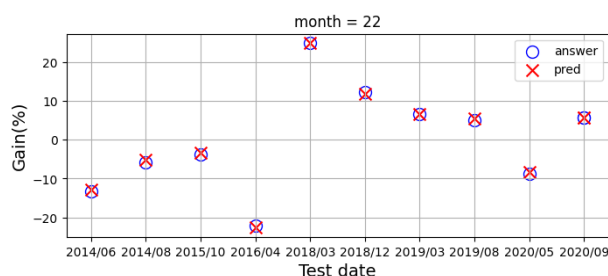


図 4 $r = 1$, $k = 22$ の推論結果

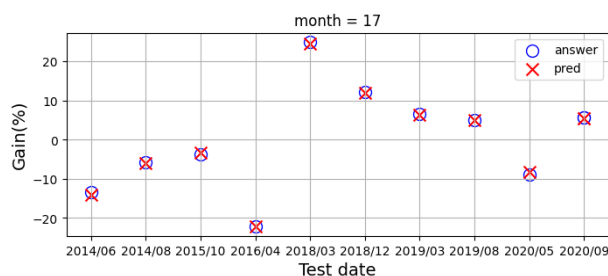


図 5 $r = 4$, $k = 17$ の推論結果

5. まとめ

本研究では，RNN を用いて次月のプレイヤーの最大同時接続数の増加率を予測する AI の実装を試みた．RNN を多層化することで，入力するデータの月の数を減らしても予測精度を損なわないことを確認した．今後の課題として，より入力するデータの月の数を減らすべく，モデルの構造等の条件を変え，実験することが挙げられる．

参考文献

- [1] Kaggle, “Popularity of games on Steam”, <https://www.kaggle.com/datasets/michau96/popularity-of-games-on-steam> (最終確認日:2024/02/03)
- [2] Steam DB, “Database of everything on Steam”, <https://steamdb.info/> (最終確認日:2024/02/03)
- [3] Steam, “Steam へようこそ”, <https://store.steampowered.com/?l=japanese> (最終確認日:2024/02/03)