LSTM の無変化性バイアスの実験的分析

石井太河 [1]　上田亮 [1]　宮尾祐介 [1]
[1] 東京大学
{taigarana, ryoryoueda, yusuke}@is.s.u-tokyo.ac.jp

概要

本研究では Long Short-Term Memory（LSTM）の学習傾向を実験的に分析する．LSTM が出力が単調な関数を学習しやすいという報告は先行研究でなされていたが，詳細な実験は行われていなかった．本研究では決定性有限オートマトン（DFA）のタイムステップに対する受理状態の変化の有無で入力記号列を 2 種に分類し，それぞれに対する LSTM の学習時間の傾向を調査する．直鎖状の DFA を学習させた結果，DFA の受理・非受理が切り替わる状態遷移の両端の状態に対応する記号列はそうでない記号列に比べ学習が遅いことが明らかになり，LSTM は出力が変化しにくい関数を学習しやすいという無変化性バイアスを持つことが支持される．

1 はじめに

本研究では，深層学習モデルの一つである Long Short-Term Memory （LSTM）[1] に対して以下の仮説 1 を立て，検証することを目的とする．

仮説 1　LSTM のモデル M は入力記号列 $s \cdot a$ と任意の記号 c に対して，末尾の記号の削除・追加で出力が変化しない，すなわち，$M(s) = M(s \cdot a) = M(s \cdot a \cdot c)$ となるような入力記号列 $s \cdot a$ を学習しやすい傾向（無変化性バイアス）がある（s は記号列，a, c は記号である）．

本研究では，末尾の記号の追加・削除による出力の変化の有無に従って入力記号列を 2 種に分類し，それぞれに対して LSTM が学習に要した時間を評価することで，上記の仮説 1 を検証する．分析を容易にするため，出力の変化は決定性有限オートマトン（deterministic finite automaton；DFA）でモデル化し，LSTM には入力を記号列，出力を受理・非受理として DFA を学習させる．なお，学習対象の DFA には図 1にあるような直鎖状のものを設定する．一般に様々なグラフ構造の DFA が無限に存在するが，受理・非受理の逐次的な変化のパターンをモデル化するには直鎖状の DFA が最もシンプル[1)]なためである．

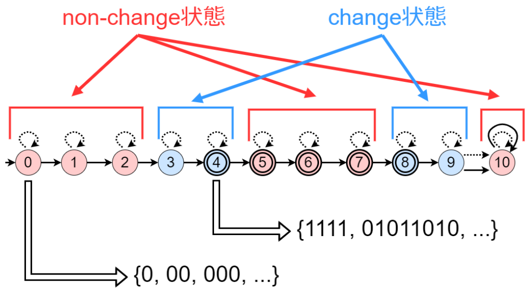

図 1　直鎖 DFA のパラメータが $(m, w) = (3, 2)$ の例．長さ $m = 3$ の non-change 状態の区間と change 状態の区間が $w = 2$ 回繰り返され，末尾に明示的に non-change 状態が付け加えられている．点線，実践はそれぞれ 0, 1 の状態遷移を表す．状態 0 には記号列 $0, 00, 000, \ldots$ が対応し，状態 4 には記号列 $1111, 01011010, \ldots$ が対応する．

図 1では入力記号列の受理・非受理の切り替わりによる分類の例が示されている．状態遷移による受理・非受理の変化がない状態 0 には記号列 $0, 00, 000, \ldots$ が対応し，変化がある状態 4 には記号列 $1111, 01011010, \ldots$ が対応する．

また，2 種の入力記号列の集合それぞれに対してモデルの学習時間を計測するにあたり，本研究では [2] に習い，勾配降下法によるモデルの学習に要したエポック数を計測した．

2 背景

2.1 本研究の位置づけ

本研究における仮説 1 は，[2] による LSTM は単調な分類関数を学習しやすいという報告を変化・無変化性に対して一般化したものである．[2] は自然言語の量化子の意味計算をオートマトンによる逐次的な処理によってモデル化し，LSTM により量化子

1）自己ループ以外のループを持たない DFA は，複数の直鎖 DFA から構成されるとみなすことができる．

の学習をシミュレートした．[2] は，モデルの学習過程においてテスト精度が安定して一定値を超えた最小エポック数を計測した結果，単調な量化子の学習はエポック数が小さくなる，すなわち学習しやすいことを報告している．ここで，分類関数 f が単調であるとは $\leq$ をブール値上の順序として，アルファベット Σ 上の任意の記号列 $s \in \Sigma^*$ と文字 $c \in \Sigma$ に対して，$f(s \cdot c) \leq f(s)$ または $f(s) \leq f(s \cdot c)$ のどちらか一方のみが成り立つことを言う．

本研究のアプローチは，$f(s), f(s \cdot c)$ 間の関係を変化 $f(s \cdot c) \neq f(s)$ と無変化 $f(s \cdot c) = f(s)$ に分解し，関数全体の単調性よりも細かく，出力ブール値の局所的な変化の有無に着目するものである．

2.2 LSTM の性質を調べることの意義

深層学習モデルは今日多くの場面で利用されており，学習の効率化や未知の挙動の抑制は重要である．LSTM は，入力記号列を逐次処理する深層学習モデルである Recurrent Neural Network（RNN）の一種であり，多くの自然言語処理タスクで使用され，分析されてきた．[3] は LSTM を含めた複数の RNN アーキテクチャについて表現能力の理論的な解析・分類を行い，LSTM がカウンタを実装可能な表現力を持つことを示した．しかし，深層学習モデルの複雑さから LSTM の全ての性質を明らかにするほど理論的解析は進んでおらず，多くの先行研究が実験的に LSTM の性質を分析している [4, 5, 6, 7, 8]．[4] は文脈自由言語を学習させることで，LSTM が再帰構造を学習し，ある程度の深さまでの汎化性能も持つことを報告している．また，[7] は文脈自由言語の学習によって LSTM の語順に対するバイアスが少ないことを示した．本研究はこれらの先行研究と同様に実験的に LSTM を分析する．

2.3 オートマトンと RNN の関係性

LSTM を含む RNN は実数ベクトルを内部状態とし，理論的には無限状態を持つが，入力記号列に対する逐次的な状態遷移はオートマトンと同様である．本研究では以下で定まる決定性有限オートマトンとの類似性を利用して LSTM の分析を試みる．

決定性有限オートマトン（deterministic finite automaton ; DFA）A は $A = (\Sigma, Q, q_0, F, \delta)$ によって定められる組である．ここで，Σ は有限アルファベット，Q は有限状態集合，$q_0 \in Q$ は初期状態，$F \subseteq Q$ は受理状態集合，$\delta : Q \times \Sigma \to Q$ は状態遷移関数である．再帰適用 δ^* は $\delta^*(q_0, \epsilon) = q_0$，$\delta^*(q_0, s \cdot c) = \delta(\delta^*(q_0, s), c)$ で定義される．さらに，記号列に対する A の分類関数 $f_A : \Sigma^* \to \{\mathrm{True}, \mathrm{False}\}$ は $s \in \Sigma^*$ に対して以下のように定められる．

$$f_A(s) \equiv \delta^*(q_0, s) \in F \tag{1}$$

オートマトンを利用した RNN の分析法として，学習済み RNN から有限オートマトンを抽出する手法が提案されている [9, 10, 11]．このような手法によって，RNN を形式的に解析することは可能となるが，これまで抽出に対する精度保証は与えられていないため，本研究では用いない．

3 手法

ここでは，本研究の分析対象である無変化性を定義し，LSTM の学習・評価に用いる手法について説明する．

3.1 無変化性

DFA $A = (\Sigma, Q, q_0, F, \delta)$ の状態 $q \in Q$ が無変化であるとは状態遷移の前後で受理・非受理が変化しないことと定義する．これは以下によって定式化される．

$$\begin{aligned} &\forall p \in Q. \forall c \in \Sigma. \\ &\quad \delta(p, c) = q \lor \delta(q, c) = p \Longrightarrow p \in F \Leftrightarrow q \in F \end{aligned} \tag{2}$$

無変化である状態を non-change 状態と呼び，無変化でない状態を change 状態と呼ぶ．non-change 状態，change 状態の集合をそれぞれ Q_n, Q_c とすると，これらに対応する記号列 S_n, S_c は $x \in \{n, c\}$ を用いて以下のように定まる．

$$S_x \equiv \{s \mid s \in \Sigma^* \land q \in Q_x \land \delta^*(q_0, s) = q\} \tag{3}$$

3.2 直鎖 DFA

記号列の受理・非受理の変化だけに注目するため，本研究ではアルファベット $\{0, 1\}$ 上の直鎖状の DFA のサブクラスのみを考慮する．本研究で扱う直鎖 DFA は，状態遷移で受理・非受理が変化しない non-change 状態の区間，受理・非受理の変化前後の change 状態の区間が交互に繰り返さた後に non-change 状態が付属するものとして定められる．このような直鎖 DFA は 2 つのパラメータ m, w によって図 1のように特徴づけられる．ここで，m は non-change 状態区間の長さ，w は non-change 状態区間と change 状態区間の繰り返しの総数である．ま

た，図 1の状態 9, 10 に見られるように，本研究で扱う直鎖 DFA の末尾の 2 状態は最小オートマトンにおいては同じ状態となるが，記号列の変化・無変化性の区別を明確化するため，本研究では明示的に最小でないオートマトンを使用する．なお，初期状態を非受理状態として固定したが，一般性は失われない[2)]．

3.3 データセット

学習データセットは記号列とその受理・非受理ラベルから構成されるが，DFA の各状態に対応する記号列の頻度差の与える影響を抑えるため，各状態に対応するデータ数が等しくなるように調整される．上記は以下のように定式化される．学習対象の DFA を $A = (\{0,1\}, Q, q_0, F, \delta)$ とする．A に対し，サイズ N の学習データセット D は記号列の最大長 L を用いて以下のように定義される．

$$S \equiv \{(s, f_A(s)) \mid s \in \{0,1\}^* \land |s| \leq \mathrm{L}\} \tag{4}$$

$$T \equiv \mathrm{choose}(S, N) \tag{5}$$

$$D \equiv \mathrm{balance}_A(T) \tag{6}$$

ここで，choose はランダムに重複なく N 個サンプリングする関数である．サンプルされたデータセットの分布を調整する $\mathrm{balance}_A$ 関数は以下の工程で定められる．

1. 各 $q \in Q$ に対して，$T_q \equiv \{s \mid s \in T \land \delta^*(q_0, s) = q\}$ を計算
2. 各 $q \in Q$ に対して，$|T_q| = \frac{N}{|Q|}$ となるように T_q を必要に応じて upsampling，downsampling したものを T'_q とする
3. 全ての T'_q を合併したものを D とする

3.4 評価指標

モデルの学習難易度を評価する指標として，[2] を参考にし，よりシンプルに，モデル出力の平均 2 値交差エントロピー誤差が閾値 t 以下になる最小のエポック MinEpoch を使用する．評価値 MinEpoch は non-change 状態と change 状態のそれぞれに対応する記号列の集合 $S_\mathrm{n}, S_\mathrm{c}$ に対して計算さる．それらは $\mathrm{MinEpoch}_\mathrm{n}, \mathrm{MinEpoch}_\mathrm{c}$ として $x \in \{n, c\}$ を用いて以下のように定められる．

$$\mathrm{MinEpoch}_x \equiv \min\{e \mid \mathrm{mean\_loss}(M_e, S_x) < \mathrm{t}\} \cup \{\mathrm{e\_max}\} \tag{7}$$

ここで，M_e はエポック e におけるモデル，mean_loss は平均 2 値交差エントロピー誤差，e_max は最大エポック数である．

4 実験設定

モデルのアーキテクチャとしては，[2] と同じく 2 層 LSTM に線形層を追加したものを使用し，最適化の際にも Adam [12] を学習率を $1.0 * 10^{-4}$ として利用する．隠れ層の次元数 hidden_size の大小比較のため，hidden_size $\in \{20, 200\}$ を使用する．一般的な設定のもとで実験を行うため，モデルのパラメータの初期化は PyTorch[3)] のデフォルト設定に準じる．実験にあたり，異なるシード値で 30 のモデルをバッチサイズ 8 で 30 エポック学習させる．

変化・無変化性を分析するにあたり，non-change 状態と change 状態の両方を持つような直鎖 DFA を学習対象とした．また，計算量の観点から，直鎖 DFA の最大状態数が 16 までのものを扱うことにした．すなわち，$\mathrm{m} \in \{1, 2, 3\}, \mathrm{w} \in \{1, 2, 3\}$ により特徴づけられる 9 通りの直鎖 DFA を使用する．データセットにおける記号列の最大長は，計算量の観点から今回扱う直鎖 DFA の最大状態数よりも 1 大きい $\mathrm{L} = 17$ とした．このときデータの総数は 262142 である．学習データサイズとしては全体数のおよそ 15% にあたる $4 * 10^4$ を用いる[4)]．また，MinEpoch を計算する際に用いる閾値としては [2] によりモデルの学習度を決定するのに用いられていた $t = 0.02$ を使用する．

5 結果と議論

5.1 LSTM は無変化性バイアスを持つ

実験の結果，以下に見るようにほとんどの設定で $\mathrm{MinEpoch}_\mathrm{c} > \mathrm{MinEpoch}_\mathrm{n}$，すなわち change 状態より non-change 状態の方が MinEpoch が高くなった[5)]．これは仮説 1 が成立し無変化性バイアスがあることを示す．また，この結果は LSTM の hidden_size によらず同様の傾向となったが，hidden_size が大きいと

2） 受理・非受理のラベルは one-hot vector で表現され，どの次元も等価であるため．

3） https://pytorch.org/

4） サンプリングしたデータに対してバランス調整を施す過程で直鎖 DFA の最大状態数程度の誤差は生じうる．

5） A に全ての実験結果が記載されている．

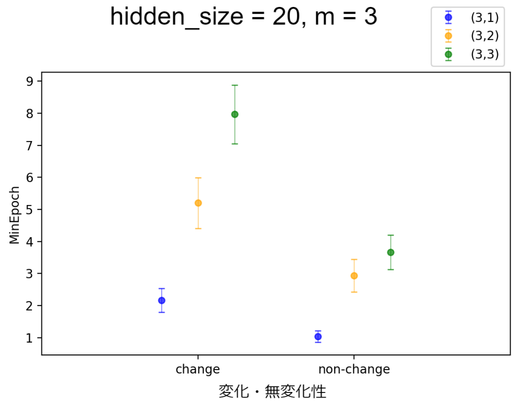

図 2　直鎖 DFA (m, w) = (3, 1), (3, 2), (3, 3) の $\mathrm{MinEpoch_c}$ と $\mathrm{MinEpoch_n}$.

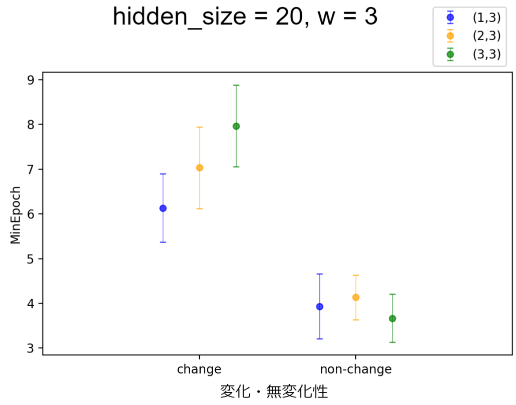

図 3　直鎖 DFA (m, w) = (1, 3), (2, 3), (3, 3) の $\mathrm{MinEpoch_c}$ と $\mathrm{MinEpoch_n}$.

そもそも学習が速く無変化性バイアスは小さくなった．以下では，分かりやすさのため hidden_size = 20 の場合の結果を軸として議論する．

5.2　出力の変化が多いと学習しにくい

図 2には change 状態区間の総数 w の異なる 3 つの直鎖 DFA について $\mathrm{MinEpoch_c}, \mathrm{MinEpoch_n}$ が示されている．w によらず $\mathrm{MinEpoch_c} > \mathrm{MinEpoch_n}$ となっており，無変化性バイアスが見られる．また，w が大きくなるほど，全体として学習が遅くなっていることも分かる[6)]．[2] では，DFA で表現される量化子の単調性が LSTM による学習しやすさの要因であると結論付けられていたが，本研究の実験結果を踏まえると，学習のしやすさの要因は LSTM の無変化性バイアスだと推測できる．実際に，非単調な DFA は単調な DFA よりも change 状態区間の数が多く，これは change 状態区間の数が多いほど全体として学習の難易度が増しているという上記の結果に符合する．

5.3　出力変化の学習には記憶が必要

図 3では non-change 状態区間の長さ m の異なる 3 つの直鎖 DFA について $\mathrm{MinEpoch_c}, \mathrm{MinEpoch_n}$ が示されている．m によらず，$\mathrm{MinEpoch_c} > \mathrm{MinEpoch_n}$ となっており，無変化性バイアスが見られる．また，m が大きくなるほど change 状態に対する学習は遅くなる傾向があるが，non-change 状態に対する学習にはあまり影響していない[7)]．今回学習対象とした直鎖 DFA は実質的に記号列中の 1 を数えるものであるため，カウントだけを考慮するならば non-change 状態と change 状態は同等なはずである．さらに，[4, 3] で議論されているように LSTM がカウンタを学習可能であることを踏まえると，change 状態間の距離としても考えられる m に対する MinEpoch の変化が change 状態の方が大きいというのは，LSTM がある程度の大きさのカウンタならばほとんど差異なく学習できる一方で，出力変化の学習については距離依存性を持ちカウンタより学習が難しいことを示唆する．

6）　ただし，$w = 1$ に関しては，$m = 1, 2$ の時は $\mathrm{MinEpoch_c}$ の分散が大きいため，この傾向が成立するかは明確ではなかった．
7）　ただし，$w = 1$ で $m = 1, 2$ のときは $\mathrm{MinEpoch_c}$ の分散が大きく change 状態の学習が遅くなる傾向は見られない．

5.4　無変化性バイアスの要因

無変化性バイアスの要因については以下のような簡単な仮説が考えられる．h_t, h_{t+1} をタイムステップ $t, t+1$ の隠れ状態ベクトル，出力層のパラメタ行列を W とすると，出力が無変化の時は $Wh_t = Wh_{t+1}$ が，変化する際には $Wh_t \perp Wh_{t+1}$ がそれぞれ理想的な場合に成立する．すなわち，無変化の際には $h_t \approx h_{t+1}$ であれば十分で，変化の際には $h_t \neq h_{t+1}$ が必要である．仮に LSTM の隠れ状態の遷移が不動点を持つように学習しやすいならば，出力が無変化である方が学習しやすくなると推測される．

6　今後の展望

本研究では，LSTM が出力変化を学習しにくいという無変化性バイアスを持つことを示した．一方，その要因については 5.4 節で議論したが，仮説に留まった．今後は，未知データに対する導出バイアスの検証実験や最適化における局所解に対する理論的な解析が必要であるだろう．

謝辞

本研究が形になる以前よりアドバイスをいただいていた鷲尾光樹氏に感謝いたします．また，実験結果の議論に参加していただいた研究室のメンバーにも感謝いたします．

参考文献

[1] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. **Neural Computation**, Vol. 9, No. 8, pp. 1735–1780, 11 1997.

[2] Shane Steinert-Threlkeld and Jakub Szymanik. Learnability and semantic universals. **Semantics and Pragmatics**, Vol. 12, No. 4, p. 1, November 2019.

[3] William Merrill, Gail Weiss, Yoav Goldberg, Roy Schwartz, Noah A. Smith, and Eran Yahav. A Formal Hierarchy of RNN Architectures. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 443–459, Online, July 2020. Association for Computational Linguistics.

[4] Jean-Phillipe Bernardy. Can Recurrent Neural Networks Learn Nested Recursion? In **Linguistic Issues in Language Technology, Volume 16, 2018**. CSLI Publications, July 2018.

[5] Abhijit Mahalunkar and John D. Kelleher. Using Regular Languages to Explore the Representational Capacity of Recurrent Neural Architectures. In Věra Kůrková, Yannis Manolopoulos, Barbara Hammer, Lazaros Iliadis, and Ilias Maglogiannis, editors, **Artificial Neural Networks and Machine Learning – ICANN 2018**, Lecture Notes in Computer Science, pp. 189–198, Cham, 2018. Springer International Publishing.

[6] Matej Makula and Ľubica Beňušková. Analysis and Visualization of the Dynamics of Recurrent Neural Networks for Symbolic Sequences Processing. In Véra Kůrková, Roman Neruda, and Jan Koutník, editors, **Artificial Neural Networks - ICANN 2008**, Lecture Notes in Computer Science, pp. 577–586, Berlin, Heidelberg, 2008. Springer.

[7] Jennifer C. White and Ryan Cotterell. Examining the Inductive Bias of Neural Language Models with Artificial Languages. In **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**, pp. 454–463, Online, August 2021. Association for Computational Linguistics.

[8] Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, and Kentaro Inui. Do Neural Models Learn Systematicity of Monotonicity Inference in Natural Language? In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 6105–6117, Online, July 2020. Association for Computational Linguistics.

[9] P. Tino and M. Koteles. Extracting finite-state representations from recurrent neural networks trained on chaotic symbolic sequences. **IEEE Transactions on Neural Networks**, Vol. 10, No. 2, pp. 284–302, March 1999.

[10] Gail Weiss, Yoav Goldberg, and Eran Yahav. Extracting Automata from Recurrent Neural Networks Using Queries and Counterexamples. In **International Conference on Machine Learning**, pp. 5247–5256. PMLR, July 2018.

[11] Takamasa Okudono, Masaki Waga, Taro Sekiyama, and Ichiro Hasuo. Weighted Automata Extraction from Recurrent Neural Networks via Regression on State Spaces. **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 34, No. 04, pp. 5306–5314, April 2020.

[12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, **3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings**, 2015.

A　全ての設定の実験結果

以下は本研究で行った全ての設定のもとでの実験結果のプロットである．

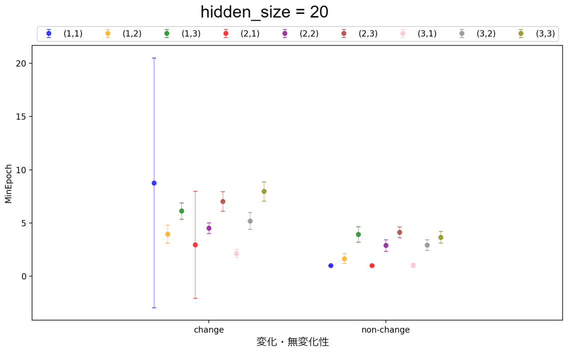

図 4　LSTM の隠れ層の次元数は hidden_size = 20 の時の直鎖 DFA のパラメータが $m \in \{1, 2, 3\}$, $w \in \{1, 2, 3\}$ の 9 通りの全ての場合の $\mathrm{MinEpoch_c}$ と $\mathrm{MinEpoch_n}$．

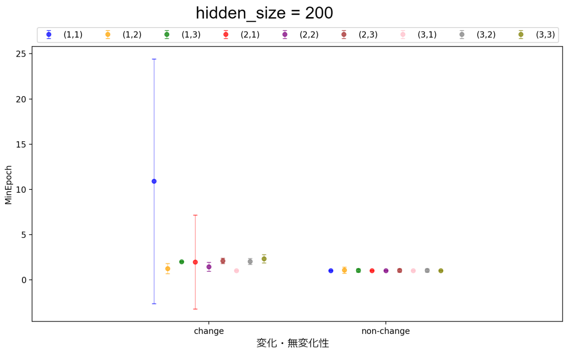

図 5　LSTM の隠れ層の次元数は hidden_size = 200 の時の直鎖 DFA のパラメータが $m \in \{1, 2, 3\}$, $w \in \{1, 2, 3\}$ の 9 通りの全ての場合の $\mathrm{MinEpoch_c}$ と $\mathrm{MinEpoch_n}$．