

深層学習による文章編集操作の時間履歴と文章評価の関係分析

藤原亮^{†1} 大場みち子^{†2} 山口琢^{†3} 高橋慈子^{†4} 小林龍生^{†5}

概要: 大学や企業、公的機関、研究機関などにおける成果発表ならびに知見の共有においては、言語、とりわけ書き言葉がコミュニケーションに用いられている。本研究では書き言葉による効果的なコミュニケーションを評価するため、深層学習により文章編集操作の時間履歴と文章評価の関係分析を行った。深層学習により文章編集操作過程を評価できれば、より客観的かつ省力的な評価手法の構築が期待できる。本研究における文章編集操作履歴は、大学のライティング演習の編集履歴からサンプリングした。文章編集履歴における、編集セルの遷移に要した時間とセルの位置を入力、ループリックにより採点された文章の総合採点を出力とし、深層学習アーキテクチャでの教師あり学習を行った。全結合ニューラルネットワーク、畳み込みニューラルネットワーク、リカレントニューラルネットワークなどのアーキテクチャを用い、まずは単体で学習を行い、その結果をもとに文章評価の分析を行う上で効果的なアーキテクチャを提案する。

キーワード: 文章編集操作、文章評価、時間履歴、深層学習

Analysis of Relation Between Text Editing-Time Process and Evaluation of Written Text by Deep Learning

RYO FUJIWARA^{†1} MICHIKO OBA^{†2} TAKU YAMAGUCHI^{†3}
SHIGEKO TAKAHASHI^{†4} TATSUO KOBAYASHI^{†5}

Abstract: Language, especially written language, is used presentation and sharing knowledge in organization such as educational institution, company, and public organization etc. In this study, to evaluate the communication by written language, we analyze the relation between text editing process, including editing time, and score of the written text by deep learning. Objective and labor-saving evaluation method is promised if deep learning can evaluate text editing process. The text editing process on this paper is sampled in writing workshop of university. The deep learning architecture learned the evaluation of text editing by the training data; the input as time to transfer editing cells and the number of the cell, and the output as the score of the written text rated by rubric. Fully-connected neural network, convolutional neural network(CNN), and recurrent neural network(RNN) are used. First each architecture evaluates the text editing process alone, then effective architectures are proposed based on the evaluation result.

Keywords: Text Editing Process, Evaluation of Written Text, Editing Time, Deep Learning

1. はじめに

人類の文明が発展した一因は言語によるコミュニケーションと言われている[1]。現代社会においても、大学や企業、公的機関、研究機関などにおける成果発表において言語、とりわけ書き言葉が用いられている。特にIT化が進んでいると言われている昨今、電子メールや業務管理システムなどにおけるコミュニケーションにおいても言葉の読み書きは重要な位置を占めている。

言語スキルを身につけるための教育工学研究では、学習過程のデータを収集・分析し、学習や教育の改善へつなげようとするラーニングアナリティクス (Learning Analytics, LA) [2]の研究が盛んである。LAの具体例として、英語リーディング過程の分析に関する研究では、英文をパラグラフごとにページ分割し、ページ遷移をログに記録すること

で1パラグラフを読むのにかかる時間を測定している事例が報告されている[3]。これらの記録したデータは学習者の行動パターンを可視化したり、成績との相関を調査したり、学習者モデルを作成するために利用されている[3][4][5]。教材に対する内容理解度を3択の問題と自由記述問題から評価している。成績との相関はこの評価と事後アンケートの関係で分析している。

我々の研究グループでは、文章編集過程を可視化するためのマトリックス型編集モデルを提案している。誰が「いつ」「どのワークシートの」「どのセルに」「何をしたか」を逐次記録する。これらの記録をもとに文章編集過程を評価する手法として、編集操作指標 (EOI) [6]、編集時間指標 (ETI) [7]による評価を提案してきた。しかしながら、これらの指標は直感的に策定されたため、採点基準に特化した編集行為により評価をつり上げられるおそれ、ならびに本

^{†1} 函館工業高等専門学校
National Institute of Technology, Hakodate College

^{†2} 公立はこだて未来大学
Future University Hakodate

^{†3} フリーランス
Independent Researcher

^{†4} 株式会社ハーティネス
Heartiness Co., Ltd.

^{†5} スコレックス
Scholex

質的に評価が高い文章の編集行為が見過ごされるおそれがある。既存の人手を要する手法では、採点者の批評眼が養われるという点もあるが、採点基準が属人的になり、採点の際に手間を要するという懸念点も否定できない。採点者に依らず採点者の手間をかけない、より客観的かつ省力的な評価方法があれば、一様な評価のもとで文章編集操作を標準化できる、評価に要する手間を節約できる、といった改善点が期待できる。

本研究では、深層学習を用いて文章編集操作の時間履歴（または時間を含まない履歴）と文章評価点数の関係を分析する。本研究では多入力で多様なデータを処理でき拡張性があるため深層学習(Deep Learning)による分析手法を採用した。本分析を通し、文章編集操作の評価に最適な深層学習のハイパーパラメータを探る。全結合ニューラルネットワーク、畳み込みニューラルネットワーク、リカレントニューラルネットワークなどの深層学習アーキテクチャを用い、まずは単体で学習を行う。その結果をもとに文章評価の分析を行う上で効果的なアーキテクチャを提案する。

2. 文章編集操作履歴データの取得

本項では文章編集操作履歴を取得したライティング授業の概要について述べる（詳細は文献[8]を参照）。

文章編集操作履歴データは、大妻女子大学の「テクニカルライティング」における受講者の操作履歴からサンプリングした。受講者は大学 2~4 年生の学生であり、計 38 名である。同授業の演習課題として、「スマートフォンとは何か」という用語説明を、60 代の携帯電話（いわゆるガラケー）しか使ったことがない人を対象に 200 字以内で説明するというものがあり、本研究では同演習における編集操作履歴データを扱った。

同授業における文章編集の際には、マトリックス型テキスト編集モデルのリファレンス実装のひとつである TopicWriter を用いた。同アプリには、図 1 に示すとおり、用語説明のための 3 つの編集セルのあるワークシートが実装されている。それぞれ、概念（それは何か、一文で簡潔に書く）、説明（具体的に説明する）、メリット・使い方について文章を書き下し、推敲を行う。演習中にはレビューの時間も設けられており、二人以上の学生がレビューを行い文章編集者に結果を返している。そのレビュー結果をもとに、文章の修正も行っている。作成された文章は、表 1 に示すルーブリックに基づいて評価が行われた。レビュー前、レビュー後、それぞれの文章について項目ごとに評価を行ったが、本研究では文章編集操作の全体を大づかみに扱うため、レビュー後の評価点数の総点にまずは着目し分析を行った。

文章編集操作履歴は Topic Writer と連動しているアプリである Writing Analysis により採取した。採取したデータは

図 2 に示すとおり csv ファイルとなっており、ある 1 つの操作について文章 ID (docId), 編集日時 (dateTime), 操作内容 (op), 編集セル番号 (id), 使用したワークシート (worksheet) などが詳細に記録されている。



図 1 「用語解説」のワークシート

Figure 1 Screenshot of worksheet for glossary.

表 1 文章評価ルーブリック

Table 1 Rubric for evaluation of text.

観点		優良	標準	要改善
構成	定義	最初の文で簡潔に用語が定義できている。	用語の定義がされているが、簡潔な一文になっていない。	用語の定義がされていない。
	具体的な説明	説明が具体的に展開されている。	説明が展開されているが、具体性がやや乏しい。	具体的な説明が書かれていない。
	メリットの記述	メリットが分かりやすく書かれている。	メリットが書かれているが、分かりやすさに改善の余地がある。	メリットが書かれていない。
表現	読み手に合わせた表現	読み手に合わせた語句と選び、分かりやすく書いている。	読み手に合わせた語句を選んでいない、または表現が 1~2 カ所あい。	読み手に合わせた語句、表現になっていない。
	一文一義	一文を 50 文字以内にし、一文一義で書いている。	一文を 50 文字以内にし、一文一義で書けていない文が 1 つある。	一文を 50 文字以内にし、一文一義で書けていない文が 2 つ以上ある。
	誤字・脱字、表現の統一	誤字・脱字がなく、表現が統一されている。	誤字脱字が 1 カ所あり、表現が統一できていない箇所が 1 カ所ある。	誤字・脱字が 3 カ所以上ある。表現が統一できていない箇所が 2 カ所以上ある。

	1	2	3	4	5	6	7	8
1	app	docId	dateTime	op	targetTypeid	vr	worksheet	
2	https://topic-writer.d1a32b73	d1a32b73	2016-10-12T01:16:58	cc_c:td	cc_24	0.7	https://matrix	
3	https://topic-writer.d1a32b73	d1a32b73	2016-10-12T01:19:13	cc_c:td	cc_24	0.7	https://matrix	
4	https://topic-writer.d1a32b73	d1a32b73	2016-10-12T01:19:24	cc_c:td	cc_25	0.7	https://matrix	
5	https://topic-writer.d1a32b73	d1a32b73	2016-10-12T01:29:22	cc_c:td	cc_25	0.7	https://matrix	
6	https://topic-writer.d1a32b73	d1a32b73	2016-10-12T01:30:06	cc_c:td	cc_25	0.7	https://matrix	
7	https://topic-writer.d1a32b73	d1a32b73	2016-10-19T00:03:03	cc_c:td	cc_26	0.7	https://matrix	
8	https://topic-writer.d1a32b73	d1a32b73	2016-10-19T00:29:09	cc_c:td	cc_26	0.7	https://matrix	
9	https://topic-writer.d1a32b73	d1a32b73	2016-10-19T01:02:42	cc_c:td	cc_24	0.7	https://matrix	
10	https://topic-writer.d1a32b73	d1a32b73	2016-10-19T01:08:24	cc_c:td	cc_25	0.7	https://matrix	
11	https://topic-writer.d1a32b73	d1a32b73	2016-10-19T01:12:39	cc_c:td	cc_26	0.7	https://matrix	
12	https://topic-writer.d1a32b73	d1a32b73	2016-10-19T01:16:32	cc_c:td	cc_26	0.7	https://matrix	
13	https://topic-writer.i22eb3285	d22eb3285	2016-10-12T01:09:29	cc_c:td	cc_24	0.7	https://matrix	

図 2 文章編集操作履歴データ

Figure 2 Text editing process data.

3. 深層学習アーキテクチャの概要

本研究では、プログラミング言語 Python のライブラリである keras を用いて文章編集履歴の評価を行う深層学習アーキテクチャを構成した[9]。機械学習のアーキテクチャを構成するにあたり、入力データと出力データの形式をテンソルで表現した。図 3 に入出力データの模式図を示す。

まずは出力データについて記述する。出力データは 20 個の成分を格納できる 1 次テンソル(20,)として記述する。総合評点は 20 点満点であり、例えば総合評点が n 点であれば第 n 成分が 1、それ以外の成分が 0 となる表現とする。

入力データは 70 個のデータを格納できる 1 次テンソル(70,)として記述する。前半部の 35 個には、編集開始時刻を基準とした編集時間 t_i を、後半部の 35 個には編集したセルの番号 c_i を格納している。編集時間を格納する際は、単位を(日)としている。例えばセルを遷移する際に 10 秒かかった際は、 $10/86,400 = 1.157 \times 10^{-4}$ を格納している。編集したセルに番号をつける際は、第 1 セルは 0.1、第 2 セルは 0.2... といった具合に 1 以下の小数を割り当てている。これらのデータ変換操作は実際に深層学習を行うにあたり過度な重みがついて学習器にバイアスがかからないようにするために行っている。なお、実際の編集データ数が 35 個を下回るときは 0 で残りのデータを埋めている。また、後述する通り学習器として畳み込みニューラルネットワーク (CNN) あるいはリカレントニューラルネットワーク (RNN) を用いる。その際は入力フォーマットの仕様上 2 次のテンソル(70,1)を用いている。さらに、比較対象として編集したセルの番号のみを格納したテンソル(35,)も用意している。

今回の編集操作履歴は計 38 名分のデータが得られたが、深層学習を行う上では十分な量とは言い難い。そこで、データ拡張として、平均 0 sec、標準偏差 5 sec の正規乱数を編集データに加えた。以上のデータ拡張を 1 名につき 1,000 回行い、元データも含め 38,038 個のデータを用いて深層学習アーキテクチャの学習を行った。実際に深層学習アーキテクチャの性能検証を行う際は、7608 個のデータで学習を行い、30430 個のデータで評価を行った。

今回の分析では様々な深層学習のアーキテクチャを用い、より効果的なアーキテクチャの構成法、つまり層の数、用いるネットワークの種類といったハイパーパラメータを探る。まずは基本的な 1 層の全結合ニューラルネットワーク (Fully-connected Neural Network), そしてこれを 2 層、3 層と層を増やした際の正答率の遷移を見る。全結合ネットワークは編集操作履歴のデータを一挙に読み込むことで学習する。次に、編集操作履歴のような、ある現象の順序を記述するシーケンスデータを扱う上で有効な手法となる畳み込みニューラルネットワーク (Convolutional Neural Network, CNN) とリカレントニューラルネットワーク (Recurrent Neural Network, RNN) でその効果を検証する。

畳み込みニューラルネットワークは、編集操作履歴のシーケンスデータの局所的なパターンをもとに学習し、リカレントニューラルネットワークはシーケンスデータの冒頭からデータを読み込み過去の履歴を記憶することにより学習する。リカレントニューラルネットワークの種類として、1 つ前の情報をもとに学習する SimpleRNN をはじめ、複数の過去データにまたがり学習する LSTM (Long Short-Term Memory), LSTM が少し効率化された GRU (Gated Recurrent Unit) が存在するので、これらについて検証する。

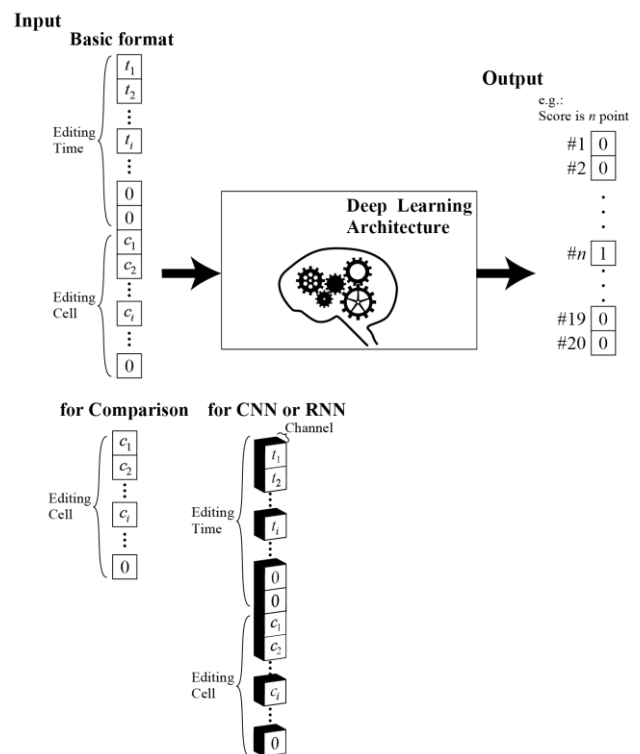


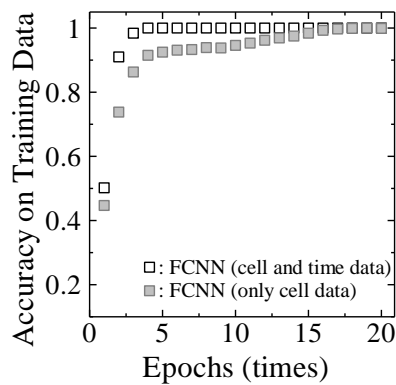
図 3 入出力データの模式図。

Figure 3 Schematic diagram of input data and output data.

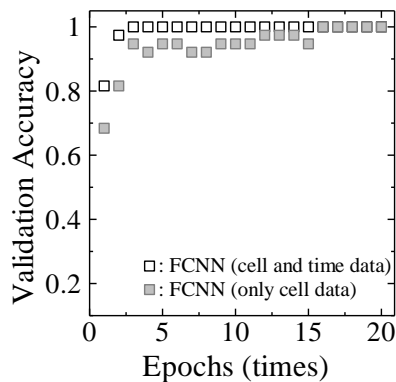
4. 深層学習による文章編集操作履歴関係分析—結果と考察

4.1 データ型による比較；時間を考慮するかしないか

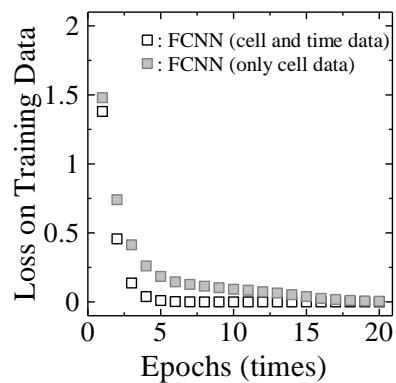
種々の深層学習アーキテクチャの比較に先立ち、データ型による学習過程の違いを述べる。入力データとして編集時間と編集セルどちらも与えたときの学習過程と、編集セルのみを与えたときの学習過程の比較を図 4 に示す。横軸は学習のエポック数であり、縦軸はそれぞれ、(a)訓練データ正答率、(b)訓練データ予測誤差 (損失)、(c)評価データ正答率、(d)訓練データ予測誤差 (損失) である。以降、同様の表記の図が続く。図 4 より、編集時間を入れたほうが正答率の上昇が早く、予測誤差も速やかに減少している。考えられる要因として、入力データ量が単純に増えたこと、ならびに離散的なセルデータに比べ情報量の多い時間データが学習の精度向上に寄与したことが挙げられる。



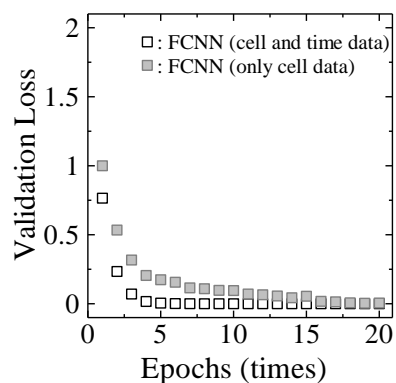
(a)



(b)



(c)



(d)

図 4 編集セル・時間と編集セルのみを与えたときの
(a)訓練データ正答率, (b)訓練データ予測誤差,
(c)評価データ正答率, (d)評価データ予測誤差.

Figure 4 Accuracy and loss for comparing input data.

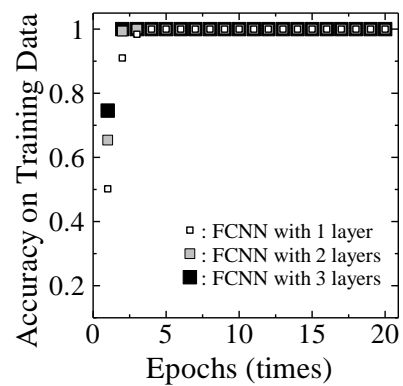
4.2 全結合ニューラルネットワークによる分析

手始めに, 全結合ニューラルネットワークにおいて層の数を換え文章編集操作履歴の関係分析を行った. 具体的なソースコードは, たとえば3層の場合は下記のとおりとなる.

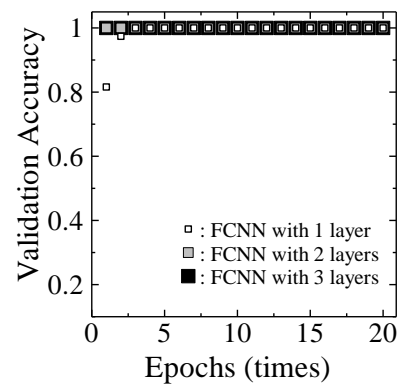
```
model_timecell.add(layers.Dense(32, activation='relu',
input_shape=timecell_val[0].shape))
model_timecell.add(layers.Dense(32, activation='relu'))
model_timecell.add(layers.Dense(32, activation='relu'))
model_timecell.add(layers.Dense(20, activation='softmax'))
```

ここで `timecell_val[0]`とは第3節で述べた編集時間と編集セルが格納された入力データの0番目を指している. 最後に `softmax` 層を入れることにより, 出力データである評点と整合するようにしている. 図5に全結合ネットワークにおける, 訓練データ正答率, 訓練データ予測誤差, 評価データ正答率, および評価データ予測誤差のエポックごとの変化を示す.

全結合ネットワークの層を多くすればするほど, 正答率が早く上昇する. これはパラメータが多くなればなるほどフィッティングの精度が上がっていると考えられる. しかし, 学習データに最適化されすぎる「過学習」という現象が生じ, 未知データに対する柔軟性を損ねてしまうおそれがある.



(a)



(b)

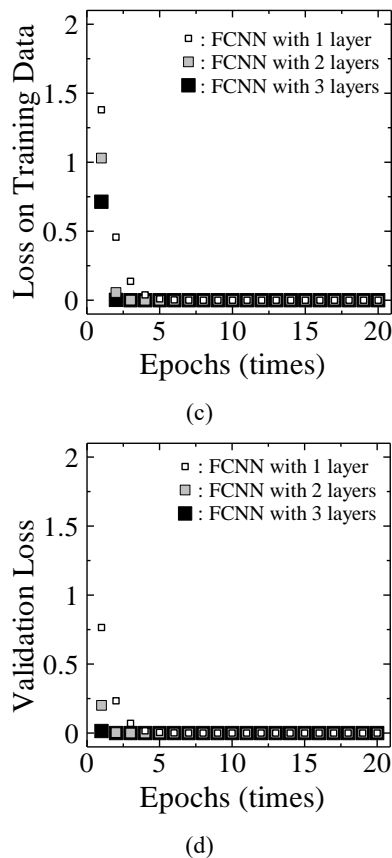


図 5 全結合 NN で層数を変えたときの

- (a) 訓練データ正答率, (b) 訓練データ予測誤差,
- (c) 評価データ正答率, (d) 評価データ予測誤差.

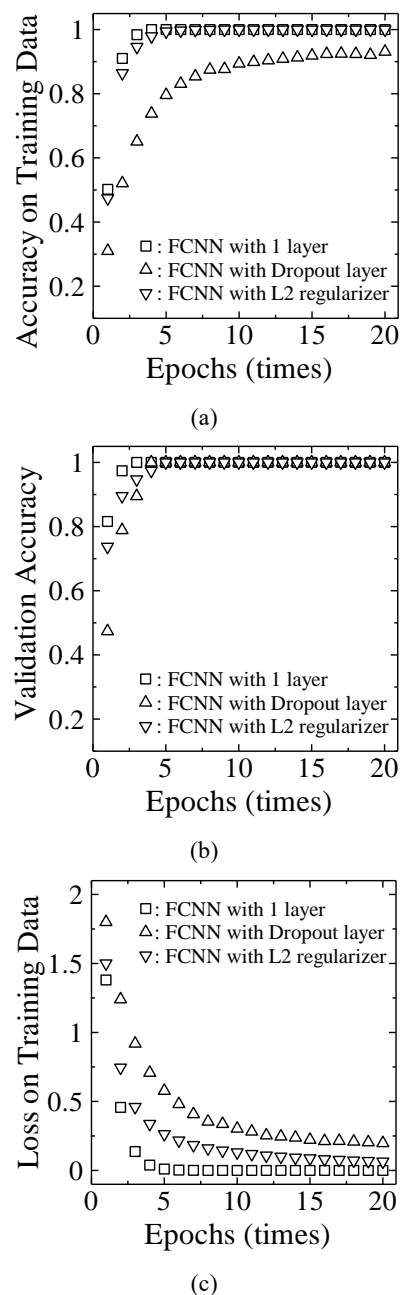
Figure 5 Accuracy and loss for comparing number of layers.

過学習を防ぐために用いられる手法として、ドロップアウト層と L2 正則化が挙げられる。ドロップアウト層とは重み付けされたノードをリセットする層であり、アーキテクチャに対し学習データに最適化しすぎた過度な学習を適宜「忘れさせる」意味合いを持つ。L2 正則化とはモデルを過度に複雑にしないための重み関数をつける措置であり、最低限の変数で入力と出力の関係を記述させるように促す働きを持つ。具体的なソースコードはそれぞれ以下のとおりである。

```
model_timecell.add(layers.Dense(32,
activation='relu', input_shape=timecell_val[0].shape))
model_timecell.add(layers.Dropout(0.5))
model_timecell.add(layers.Dense(20, activation='softmax'))
```

```
model_timecell.add(layers.Dense(32,
kernel_regularizer=regularizers.l2(0.001),
activation='relu', input_shape=timecell_val[0].shape))
model_timecell.add(layers.Dense(20, activation='softmax'))
```

1 層の全結合ネットワークに対し、ドロップアウト層を追加したアーキテクチャ、L2 正則化を施したアーキテクチャの訓練データ正答率、訓練データ予測誤差、評価データ正答率、および評価データ予測誤差のエポックごとの変化を図 6 に示す。ドロップアウト層を追加することにより学習速度に鈍りが見られていることから、この層における効果が適切に現れていると考えられる。L2 正則化を施してもあまり変わらなかった。これは入力データ量とアーキテクチャのノード数がともに少なく、入出力データ量に対し適切なパラメータで関係を記述できているからと考えられる。よりノード数を増やすことにより、L2 正則化が有効となる可能性もある。



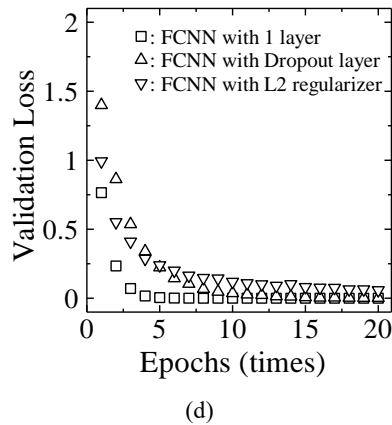


図 6 過学習を防ぐ手法を実装したときの
(a)訓練データ正答率, (b)訓練データ予測誤差,
(c)評価データ正答率, (d)評価データ予測誤差.

Figure 6 Accuracy and loss for dropout and L2 regularizer.

4.3 畳み込みニューラルネットワークによる分析

次に畳み込みニューラルネットワーク (CNN) の性能を評価する. 畳み込みニューラルネットワークとは, 編集操作履歴のシーケンスデータの局所的なパターンをもとに学習するアーキテクチャであり, 今回は編集回数にして5回分のパターンを時間および編集セルをそれぞれ認識した上で, これらのパターンのセットについて重み付けを行う. 具体的なソースコードは以下のとおりである.

```
timecell_train = timecell_train.reshape(7608, 70, 1)
timecell_val = timecell_val.reshape(30430, 70, 1)
model_timecell.add(layers.Conv1D(32, 5,
activation='relu', input_shape=(70, 1)))
model_timecell.add(layers.GlobalMaxPooling1D())
model_timecell.add(layers.Dense(20, activation='softmax'))
```

最初の 1, 2 行目では CNN の入力データのフォーマットに整合するように reshape を行っている. 最後の softmax 層に入る前に最大値プーリング演算 MaxPooling を施しダウンサンプリングしている.

図 7 より, 全結合ネットワークよりも学習曲線に鈍りが見られた. それは, それぞれの局所的なパターンを認識した上で学習を行っているからであり, すべてのデータを一括で学習する全結合ネットワークに比べ学習に鈍りが生じるのは妥当であると考えられる. 全結合ネットワークよりも, 学習曲線の上で性能が悪いながらも, 局所的なパターンを認識できるのは本アーキテクチャの強みであると考えられる. なぜなら, 実際の運用にあたり編集操作を総体的に判断して評価を下すのは, 文章作成者に対する示唆としてはリアルタイム性を欠くためである.

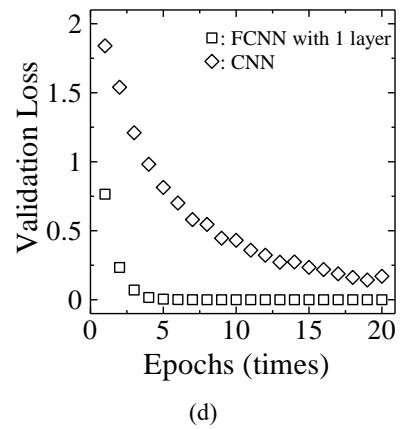
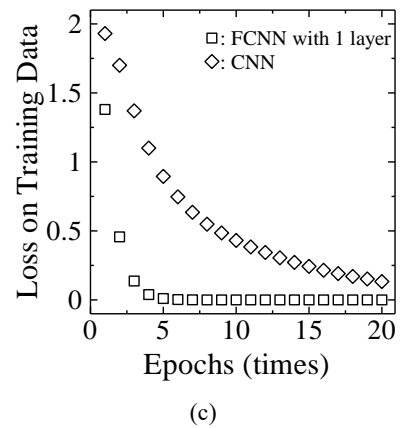
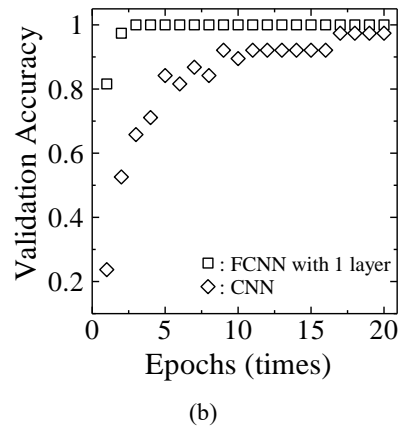
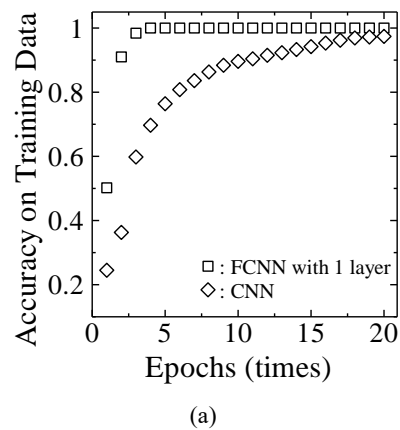


図 7 CNN を実装したときの
(a)訓練データ正答率, (b)訓練データ予測誤差,
(c)評価データ正答率, (d)評価データ予測誤差.

Figure 7 Accuracy and loss for convolutional neural network.

4.4 リカレントニューラルネットワークによる分析

リカレントニューラルネットワーク (RNN) の性能を評価する。リカレントニューラルネットワークはシーケンスデータの冒頭からデータを読み込み過去の履歴を記憶することにより学習する。リカレントニューラルネットワークの種類として、1つ前の情報をもとに学習する SimpleRNNをはじめ、複数の過去データにまたがり学習する LSTM (Long Short-Term Memory)、LSTM が少し効率化された GRU (Gated Recurrent Unit) が存在する。たとえば SimpleRNN を実装する際の具体的なソースコードは以下のとおりである。

```
timecell_train = timecell_train.reshape(7608, 70, 1)
timecell_val = timecell_val.reshape(30430, 70, 1)
model_timecell.add(SimpleRNN(32, input_shape=(70, 1)))
model_timecell.add(layers.Dense(20, activation='softmax'))
```

図 8 より、学習曲線に鈍りやばらつきが見られる。それは、数個前のシーケンスを記憶する RNN の性質に由来するものと考えられる。これも前項と同様、全結合ネットワークよりも、学習曲線の上で性能が悪いながらも、局所的な履歴を認識できるのはリアルタイム性の観点から見て本アーキテクチャの強みであると考えられる。

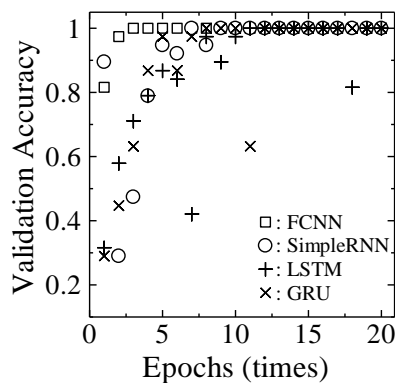
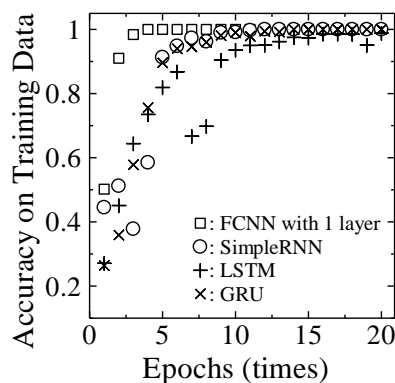
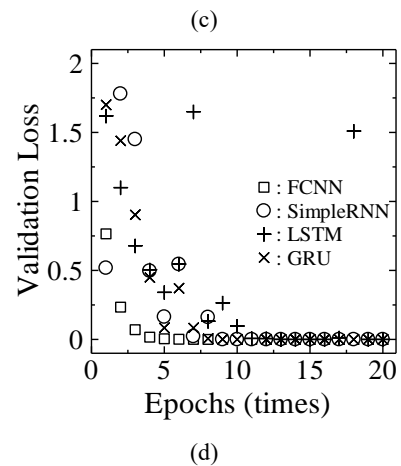
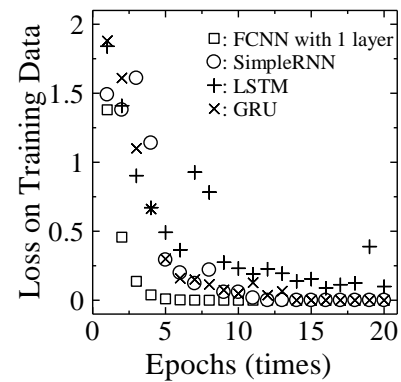


図 8 RNN を実装したときの
(a)訓練データ正答率, (b)訓練データ予測誤差,
(c)評価データ正答率, (d)評価データ予測誤差。

Figure 8 Accuracy and loss for recurrent neural network.

4.5 CNN と RNN を組み合わせたアーキテクチャによる分析

最後に、今まで述べてきた個別の深層学習アーキテクチャの分析結果をもとに、リアルタイム性に優れ、学習曲線の上で性能の良いアーキテクチャを提案する。具体的なソースコードは以下の通り、CNN と RNN を組み合わせたものとなる。

```
timecell_train = timecell_train.reshape(7608, 70, 1)
timecell_val = timecell_val.reshape(30430, 70, 1)
model_timecell.add(layers.Conv1D(32, 5, activation='relu'))
model_timecell.add(layers.GlobalMaxPooling1D())
model_timecell.add(layers.Reshape((32,1)))
model_timecell.add(SimpleRNN(32, input_shape=(70, 1)))
model_timecell.add(layers.Dense(20, activation='softmax'))
```

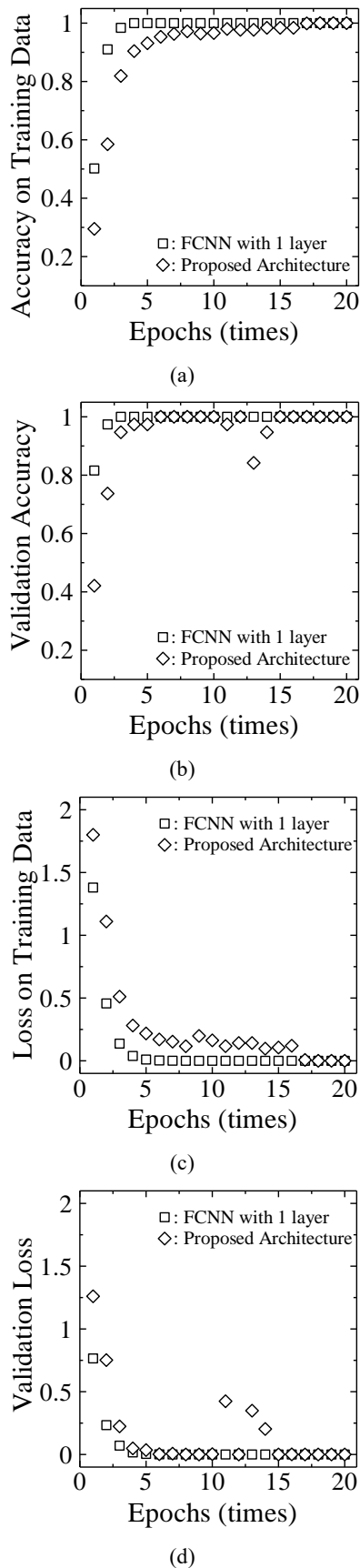


図 9 CNN と RNN を組み合わせたときの
(a)訓練データ正答率, (b)訓練データ予測誤差,
(c)評価データ正答率, (d)評価データ予測誤差.
Figure 9 Accuracy and loss for proposed architecture.

図 9 より, 正答率・予測誤差ともに収束が早く, かつ編集操作履歴のパターンを用いるためリアルタイム性に優れるため, 実用面においても使用に耐えるものと考えられる. 更にアーキテクチャを組み合わせることも想定できるが, パラメータの過多による過学習を防ぎ, 未知データに対する柔軟性を確保する措置も取る必要があると予想される.

5. おわりに

本稿では, 深層学習を用いて文章編集操作の時間履歴 (または時間を含まない履歴) と文章評価点数の関係を分析した. 本研究では多入力で多様なデータを処理でき拡張性があるため深層学習(Deep Learning)による分析手法を採用した. 全結合ニューラルネットワーク, 畳み込みニューラルネットワーク, リカレントニューラルネットワークなどの深層学習アーキテクチャを用い, まずは単体で学習を行った. その結果をもとに文章評価の分析を行う上で効果的なアーキテクチャを提案した.

今後の課題としては以下の点が挙げられる

- テキストエディタのバックエンドとしての深層学習アーキテクチャの実装
- データ拡張について
 - 標準偏差 5 秒の正規乱数は妥当なのかどうか, 他に妥当なデータ拡張手法はないか調査していきたい.
- 評価について
 - 今回は特定のルーブリックをもとに採点を行ったが, 様々な観点での評価結果を加味した評点を学習させることにより, より客観性が増すものと期待される.
 - 実際の運用においては, 定期的に人的な評価を行い本アーキテクチャに学習させ, 評価基準の更新を行い, 評価基準の偏りを防ぐ措置をとる必要があると予想される.
 - 評価の粒度を詳細に深くすることに努めたい.
 - ◇ 今回は深層学習のアーキテクチャについて重点的に考えるため, 出力データを簡単に総合評点としたが, ルーブリックの項目ごとの点数をひとまとめにしたテンソルにした際についても機械学習のアーキテクチャを構築したい.
 - ◇ また, レビューを受けてから修正する過程も含めて分析したので, レビュー前, レビュー後それぞれの評価をもとにした分析も行っていきたい.

以上の課題は残るものの, 深層学習を用いた客観的かつ省力的な文章編集履歴の評価手法について, その基礎を形作ることができたと言えるであろう. 今後, 本稿で提案した評価手法のより良い運用, および改善を図っていく.

謝辞 本研究は JSPS 科研費 17K01085 の助成を受けたものです。

参考文献

- [1] ユヴァル・ノア・ハラリ, 柴田裕之 (訳) 『サピエンス全史 (上)文明の構造と人類の幸福』, 河出書房新社, 2016.
- [2] 緒方広明, 殷成久, 大井京, 大久保文哉, 島田敬士, 小島健太郎, 山田政寛, 大学におけるラーニングアナリティクスに基づく授業改善と教育革新, 電子情報通信学会総合大会, TK-10-4, 2016.
- [3] 荒本道隆, 佐藤一裕, 中原歌織, 平沢茂一, 詳細な学習ログを用いた英語リーディング過程の分析～(1)リーディング過程の可視化, 情報処理学会第 77 回全国大会, 4G-03, 2015.
- [4] 中野美知子, 吉田諭史, 須子統太, 玉木欽也, ギエルモ エンリケズ, 詳細な学習ログを用いた英語リーディング過程の分析～(2) ログデータから見た成績との関係, 情報処理学会第 77 回全国大会, 4G-04, 2015.
- [5] 中澤真, 梅澤克之, 小林学, 小泉大城, 後藤正幸, 平沢茂一, 詳細な学習ログを用いた英語リーディング過程の分析～(3) リーディング過程における学習者モデル, 情報処理学会第 77 回全国大会, 4G-05, 2015.
- [6] 大場みち子, 山口琢, 高橋慈子, 小林龍生, 論理的文章作成における文章評価と編集操作との関係分析, 情報処理学会情報シンポジウム(SSS2016), 3-3, 2016.
- [7] 藤原亮, 大場みち子, 山口琢, 高橋慈子, 小林龍生, 時間を考慮した文章編集操作と文章評価の関係分析, 情報処理学会研究報告コンピュータと教育 (CE), 2018-CE-144, 1-6, 2018.
- [8] 大場みち子, 山口琢, 高橋慈子, 小林龍生, 文章作成における文章評価と編集操作との関係分析, 情報処理学会研究報告コンピュータと教育 (CE), 2016-CE-137, 1-8, 2016.
- [9] François Chollet, 巢籠悠輔 (監訳), 株式会社クイープ(訳) 『Python と Keras によるディープラーニング』, 株式会社マイナビ出版, 2018.