

パズルを利用したプログラミング学習教材における 操作パターンの特徴抽出

藤原亮¹ 山口琢² 大場みち子³

概要：本研究では、プログラミングコードの編集過程におけるアルゴリズム的思考を養うためのツールである「ジグソー・コード」における操作パターンの特徴抽出を試みる。操作パターンの特徴を抽出することにより、学習者に共通の思考過程を理解でき、効果的な教材の作成が期待できる。ジグソー・コードとは、行ごとにランダムに並べられたプログラムコードを並び替える形式のプログラミング学習のための web アプリケーションである。深層学習等のアルゴリズムを用いて編集ログデータを解析し、学習者に共通の操作パターンを抽出し、操作中の思考過程を考察する。

キーワード：プログラミング教材，思考過程，操作ログ，深層学習，ジグソー・コード

Feature Extraction of Editorial Log in Programming Education Tool with Puzzle-Like Operation

RYO FUJIWARA^{†1} TAKU YAMAGUCHI^{†2}
MICHIKO OBA^{†3}

Abstract: In this study, we aim to extract the features of editorial log in “Jigsaw Code”, the tool for developing algorithmic thinking in edition of programming code. By extracting the editorial features, we can understand thinking process of the learner, and can develop effective educational tools. “Jigsaw Code” is the web application with the programming codes of which line is placed at random. The learner rearranges the random codes. We use deep learning to analyze the relation between the editorial log and the result, extract the common features of the editorial log, and discuss the thinking process in edition of programming code.

Keywords: Programming Instruction Tool, Thinking Process, Editorial Log, Deep Learning, Jigsaw Code

1. はじめに

昨今 IT 革命と言われている社会の動きは産業革命と相似形の関連があると言われている。IT 革命と同時に相当数の新産業が生まれ、コンピュータの出現によるものよりもはるかに大きな技術の変化、産業構造の変化、経済構造の変化、さらには社会構造の変化が見られるようになる[1]。

このような社会情勢の中で、プログラミングスキルは殊更に重要視されている。その上、初等教育におけるプログラミングの必修化に伴い、思考力と共にプログラミング教育への関心が高まっている[2][3]。プログラミング教育において、学習者への評価方法はテスト結果などのアウトプットでの評価が主流であり、プログラミング時の過程や思考などは評価していない。プログラミング時の思考過程を示すことができれば、学習者の傾向や躓きの要因を把握でき、適切な指導が期待できる。しかしながら、プログラミングは論理的思考力、言語力、発想力、数学力など、多様な技能を同時に要求しているため、プログラミングを不得手とする学習者にはどのような技能がどれだけ不足しているのかが不明瞭になるとも言われている[4]。

多様な能力を要し、思考過程の明示化が困難であるプログラミングのスキルを向上させるため、現在までに種々の試みが行われてきた。Parson は、Parson’s Puzzle という与えられたフローチャートに対応するコードを並べるパズル式のプログラミング教材を開発した[5]。また、問題文およびその説明に対応するプログラミングコードの書かれたカードを解答欄に並べる、カード方式プログラミング教材も開発されている[6][7]。以上の Parson’s Puzzle およびカード方式プログラミング教材において、様々な方法で学習者の評価が行われてきた。Parson は、チューターによる学習者のパフォーマンスの目視確認を行わせ、受講者のパズルの改善点等に関するコメントを集めた[5]。Ihantola からも同様に、彼らの開発した Two-Dimensional Parson’s Puzzles において学習者の観察を行い、思考発話法 (Think Aloud Protocol) により学習者が回答中にどのような思考をしているのかを調査した[8]。この際、議論中の雰囲気オープンするため、ビデオやレコーダーによる映像や音声の記録は行っていない。また、カード方式プログラミング教材において、一般的なコーディング演習方式とカード演習方式の認知負荷を評価するアンケートを行った事例もある[4][9]。以上の手法

1 函館工業高等専門学校
National Institute of Technology, Hakodate College
2 フリーランス
Independent Researcher

3 公立はこだて未来大学
Future University Hakodate

は定性評価を行うためのいわゆるアナログ式の評価手法であるが、学習ログデータを用いた定量的な評価も行われている。Kumarらは、Parson's Puzzleを解く際のログデータ解析を通し、バックス・ナウア記法(BNF)によるパズル解答戦略の記述法を提案し、受講者の戦略を定量評価した[10]。森永らは、カード方式プログラミング教材において、レーベンシュタイン距離という尺度で学習者の答えと正解との距離を計算し、その時系列変化を評価した[7]。

我々の研究グループにおいても同様に、プログラミング時における思考および問題解決過程を分析するために様々な試みを行ってきた[11][12][13]。我々は「ジグソー・コード」という、行ごとにランダムに並べられたプログラムコードを並び替える形式のプログラミング学習のためのwebアプリケーションを用いている。Parson's Puzzleやカード方式プログラミング教材と本質的に異なるところは、操作した行や時間など詳細な操作ログを収集することにより、プログラミング時における思考および問題解決過程を分析できることである。本手法を用いることにより、今まで行われてこなかった詳細な操作ログに基づく学習者の思考および問題解決過程の分析が可能となる。

本研究では、プログラミングコードの編集過程におけるアルゴリズム思考を養うためのツールである「ジグソー・コード」における操作パターンの特徴抽出を試みる。操作パターンの特徴を抽出することにより、学習者に共通の思考過程を理解でき、効果的な教材の作成が期待できる。深層学習を用いて編集ログデータを解析し、学習者に共通の操作パターンを抽出し、操作中の思考過程を考察する。分析手法として、深層学習による操作ログと解答の紐づけを行い、学習したアーキテクチャの特定のフィルタを活性化させる入力を得、その入力を読むことにより操作を行う上でクリティカルなパターンを求める。

2. 実験方法

実験対象者は、公立はこだて未来大学の学部2年生対象講義「情報処理演習I」を履修している学生のうち、情報提供の同意が得られた48名(情報システムコース8名、情報デザインコース8名、知能システムコース32名)である。「情報処理演習I」はJava言語を題材とした課題を通して、ソフトウェア開発プロセスにおける基本技能を習得することを目的とした講義である。

実験手順は次のとおりである：

1. 概要説明：初回の実験前に、実験の概要やジグソー・コードの操作方法を説明する。
2. 実験問題の回答：用意した実験問題3問を、各回の授業後に回答する。
3. 解答解説の提示：回答後に実験問題3問に対応した解答解説を提示する。

実験は全9回を8週に分けて、授業の復習として実施した。

実験回ごとの単元概要を表1に示す。実験問題はすべてJavaのプログラムコードとし、「情報処理演習I」の各授業回に対応した範囲の問題を作問して出題した。作問時には、並べ替えの対象行を5行以上とした。これは少ない行をランダムに並べ替えた際に、正解の並びになる確率を下げためである。作問ではインデントによる区別やコメントによる順番指定、同一処理の排除などにより順不同の回答をできるだけ減らす工夫をした。これらは正解のパターンが多くなると分析が困難になるからである。復習として受講者の理解に役立てるために、出題内容についての解答解説を作成した。本稿では、以上の授業のうち人数も多くデータのバリエーションが比較的豊富な第1回の授業に関する操作ログデータを解析する。

表1 実験回ごとの単元概要

Table 1 Theme of class.

| 回 | 単元概要 |
|-----|-------------------------------------|
| 第1回 | if文, switch文 |
| 第2回 | 配列, 拡張for文, メソッド |
| 第3回 | オブジェクト指向, getterメソッド, setterメソッド |
| 第4回 | オブジェクト指向, メソッドの引数と返り値 |
| 第5回 | オブジェクト指向, メソッドの引数と返り値 |
| 第6回 | オブジェクト指向, インスタンスのメソッド呼び出し |
| 第7回 | オブジェクト指向, ArrayList, 引数の指定方法 |
| 第8回 | オブジェクト指向, クラスの継承 |
| 第9回 | HashMap, 外部ファイルの読み書き |

本研究で扱う教材アプリのジグソー・コードとは、行ごとにランダムに並べられたプログラムコードを並び替える形式のプログラミング学習のためのwebアプリケーションである。図1はパズル問題(JavaScriptのプログラム)の例である。パズルのピースには“s1”等の番号が振られたIDがあり、測定・分析に用いられる。パズルを開始すると、これらの行がランダムに並び替えられてプレイヤーに提示される。プレイヤーはドラッグ&ドロップで行を操作することにより並び替え、適切と考える順序になったところで完成ボタンを押す。ジグソー・コードではプレイヤーの操作を記録している。操作ログの生データを図2に示す。「docid」はパズル自体のIDであり、操作状況の識別に用いる。

「dateTime」は操作を行った日時であり、最終列の「dateTimeMS」ではミリ秒単位で記録している。「op」は操作内容であり、開始(cc_start)・ドラッグ(cc_drag)・ドロップ

(cc_drop)・終了(cc_completed)のいずれかが入る。「id」は操作対象である行の ID である。「prev」は操作対象の前の行の ID である。「next」は操作対象の後の行の ID である。「order」は操作された瞬間の行の配列である。

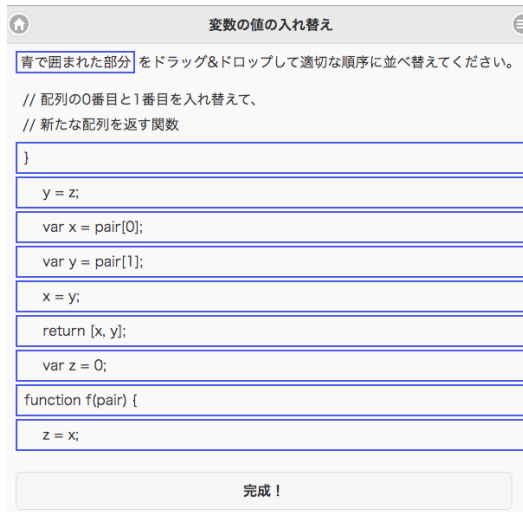


図 1 パズル問題 (JavaScript のプログラム) の例
Figure 1 Example of puzzle question (JavaScript Program).

| A | B | C | D | E | F | G | H | I | J | K | L | M |
|----|-------------|-------------|-------------|------|------------|-----|------|------|-------------|-------------------------------------|-----------|---------|
| 1 | css | board | colTime | cp | targetType | id | prev | next | order | url | worksheet | dateYMS |
| 2 | https://cc- | 2019051405- | 2019-05-14T | drop | span | | | | | https://cc-2019-05-14T05:20:30.405Z | | |
| 3 | https://cc- | 2019051405- | 2019-05-14T | drop | span | i10 | i11 | i12 | i10.i11.i12 | https://cc-2019-05-14T05:21:05.943Z | | |
| 4 | https://cc- | 2019051405- | 2019-05-14T | drop | span | i10 | i15 | i13 | i10.i15.i13 | https://cc-2019-05-14T05:21:08.832Z | | |
| 5 | https://cc- | 2019051405- | 2019-05-14T | drop | span | i10 | i15 | i13 | i10.i15.i13 | https://cc-2019-05-14T05:21:09.317Z | | |
| 6 | https://cc- | 2019051405- | 2019-05-14T | drop | span | i10 | i15 | i13 | i10.i15.i13 | https://cc-2019-05-14T05:21:09.981Z | | |
| 7 | https://cc- | 2019051405- | 2019-05-14T | drop | span | i11 | i13 | i12 | i10.i15.i13 | https://cc-2019-05-14T05:21:10.395Z | | |
| 8 | https://cc- | 2019051405- | 2019-05-14T | drop | span | i11 | i10 | i15 | i10.i11.i15 | https://cc-2019-05-14T05:21:17.878Z | | |
| 9 | https://cc- | 2019051405- | 2019-05-14T | drop | span | i14 | i17 | | i10.i11.i15 | https://cc-2019-05-14T05:21:18.118Z | | |
| 10 | https://cc- | 2019051405- | 2019-05-14T | drop | span | i14 | i13 | i12 | i10.i11.i15 | https://cc-2019-05-14T05:21:17.468Z | | |
| 11 | https://cc- | 2019051405- | 2019-05-14T | drop | span | i10 | i16 | i17 | i10.i11.i15 | https://cc-2019-05-14T05:22:08.468Z | | |
| 12 | https://cc- | 2019051405- | 2019-05-14T | drop | span | i10 | i12 | i18 | i10.i11.i15 | https://cc-2019-05-14T05:22:09.512Z | | |
| 13 | https://cc- | 2019051405- | 2019-05-14T | drop | span | i10 | i16 | i17 | i10.i11.i15 | https://cc-2019-05-14T05:22:09.711Z | | |
| 14 | https://cc- | 2019051405- | 2019-05-14T | drop | span | i10 | i16 | i17 | i10.i11.i15 | https://cc-2019-05-14T05:22:09.711Z | | |
| 15 | https://cc- | 2019051405- | 2019-05-14T | drop | span | i10 | i16 | i17 | i10.i11.i15 | https://cc-2019-05-14T05:22:10.852Z | | |
| 16 | https://cc- | 2019051405- | 2019-05-14T | drop | span | i10 | i17 | | i10.i11.i15 | https://cc-2019-05-14T05:22:10.852Z | | |
| 17 | https://cc- | 2019051405- | 2019-05-14T | drop | span | i12 | i14 | i16 | i10.i11.i15 | https://cc-2019-05-14T05:22:27.088Z | | |
| 18 | https://cc- | 2019051405- | 2019-05-14T | drop | span | i12 | i11 | i15 | i10.i11.i15 | https://cc-2019-05-14T05:22:28.421Z | | |

図 2 操作ログの生データ
Figure 2 Raw data of editorial log.

3. 解析対象となる操作履歴データの構造

入力データは 2 次元テンソル(90,6)として記述する。一学生あたりの平均の操作ログ数は最少 6 回, 最多 34 回, 平均 17.67 回のため, 今回のケースでは操作ログを格納する上では十分な容量と判断した。操作の一連の流れとしては, パズルの開始, 特定行のドラッグ, ドロップ, … (以下ドラッグとドロップの繰り返し) …, そしてパズルの終了となる。この一連の流れを以下の入力データ構造として時空間的に格納した。第 1 成分は操作回数を表し, 第 2 成分はそれぞれ「操作対象の行 (図 3 の i_m)」「操作対象の前にあった行 (同 p_{pm})」「操作対象の後にあった行 (同 n_{pm})」「操作後に操作対象の前に来る行 (同 p_{tm})」「操作後に操作対象の後に来る行 (同 n_{tm})」「前回の操作からの経過時間 (同 t_m)」となっている。意図としては, 操作者の注意が「操作対象の行」およびその前後のみに注意を向けていると仮定し,

その注意の履歴を読み解くことで操作者の思考パターンの解明に寄与すると考えた。出力データは 1 次元テンソル(10,)として記述する。テンソルの内容は「最終的に得られた行の配列 (図 3 の o_m)」を表している。

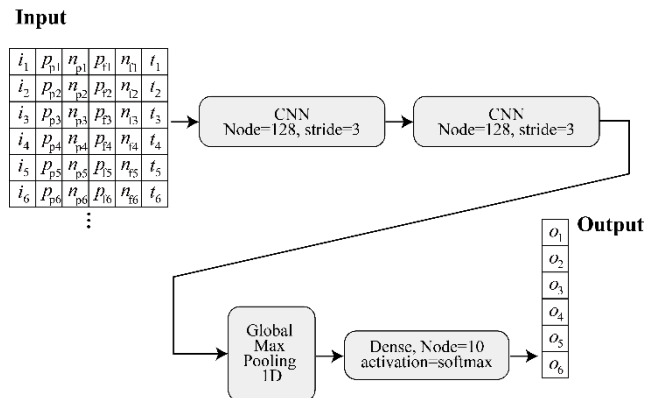


図 3 操作履歴データ構造と深層学習アーキテクチャ
Figure 3 Structure of editorial log data and architecture of deep learning.

4. 深層学習アーキテクチャの構成

深層学習によりジグソー・コードの操作過程を分析した。深層学習アーキテクチャは, Python のライブラリである keras を用いて構成した。深層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と深層学習アーキテクチャの構成を示す。深層学習のアーキテクチャは, 編集履歴データの流れを時空間的に把握するため, 畳み込みニューラルネットワーク (Convolutional Neural Network, 以下 CNN) レイヤを 2 層用いた。CNN のノード数は 128, ストライド数は 3 とした。

学習を行うに当たり豊富なデータを得るために, データ拡張を行った。「前回の操作からの経過時間」に対して正規乱数を用いた係数を掛け, 標準偏差にして 10%, 平均にして 0%の確率分布となるようにした。データ数ははじめ 48 であったが, 拡張をすることで 4848 となった。また, データの順番はランダムに並べ替えた。また, 今回は単純のため, 本来は 2 次元配列にあるデータを 1 次元配列に格納し直して学習を行った。後に述べる考察を行う際, 2 次元配列に格納し直した。以上のデータを k 分割交差検証により繰り返し学習した。今回は k=4 とし, はじめの 3 クォータを訓練および評価データ, 最後の 1 クォータをテストデータとした。

5. 操作履歴データの解析と考察

学習したアーキテクチャを解析することにより, 学生の操作のうちどのようなパターンが結果に作用しているかを

考察する。学習したアーキテクチャの解析には、勾配上昇法による CNN フィルタの可視化手法を用いた。勾配上昇法により、ある特定の CNN フィルタを活性化させる入力を得、その入力を読むことによりジグソー・コードを解く上でクリティカルな操作パターンを求める。

図 4 に第 1 層 0 番目のフィルタを活性化させる入力を、カラーグラデーションで示す。図 4 より、操作者の思考の 1 パターンとして「序盤はパズルの盤面をいじることで感触をつかみ」「中盤・終盤は黙考してパズルの盤面を調整している」というパターンを読み取ることができる。

以下、図 4 より詳細を解説する。第 1 層 0 番目のフィルタを活性化させる入力の序盤のうち、行の配置のある箇所が活性化している。これは 3 回目の行のドラッグおよびドロップ操作に対応しており、盤面を読む動作に対応していると捉えられる。第 1 層 0 番目のフィルタを活性化させる入力の中盤から終盤にかけて、時間の格納されているセルが活性化している。これは約 15 回目の行のドラッグおよびドロップ操作前の、操作待機時間に対応している。授業中にジグソー・テキストを操作したことを鑑みると、過去に操作してきた盤面を眺め思考している時間に対応していると捉えることができる。同様に、第 1 層 0 番目のフィルタを活性化させる入力の終盤のうち、時間の格納されているセルが活性化している。これは約 30 回目の行のドラッグおよびドロップ操作前の、操作待機時間に対応している。他のフィルタの活性化も分析してみる。図 5 に第 1 層 21 番目のフィルタを活性化させる入力を示す。第 1 層 21 番目のフィルタを活性化させる入力では、序盤は同様に配置のある場所が活性化しており、盤面を読む動作をしていると捉えられる。0 番目のフィルタと異なる点として、中盤および終盤（約 15 回目の行のドラッグおよびドロップ操作）でも同様に配置のある場所が活性化しており盤面を読んでいることが分かる。終盤は 0 番目のフィルタと同様に、時間の格納されているセルが活性化しており、過去に操作してきた盤面を眺め思考していると捉えられる動作が確認できる。

最後に、0 番目から 63 番目まですべての第 1 層フィルタを活性化させる入力に対し、相加平均をとったものを図 6 に示す。これはすべての入力の平均をとったためコントラストは落ちているものの、序盤はパズルの盤面をいじることで感触をつかみ、終盤はへ行くに従い黙考してパズルの盤面を調整している様子が伺える。

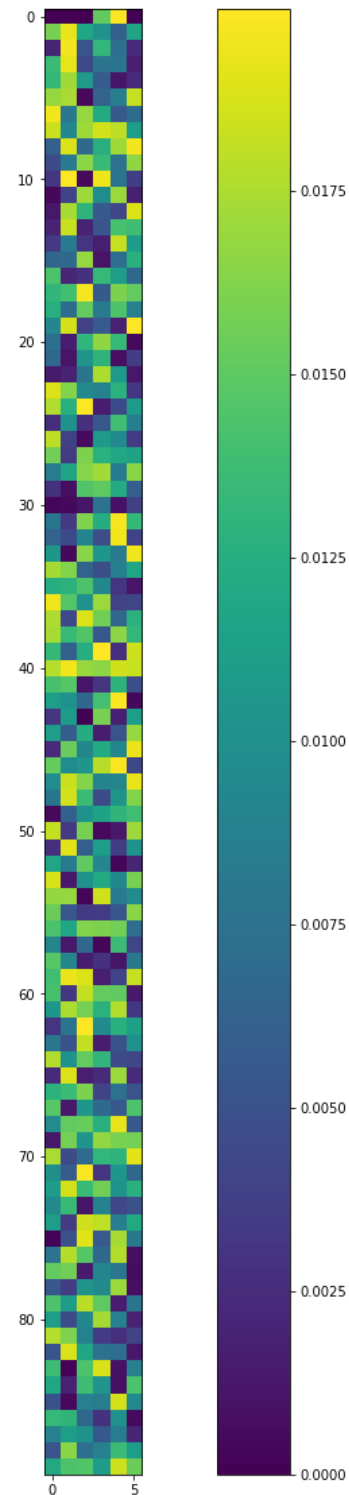


図 4 第 1 層 0 番目の CNN フィルタを活性化させる入力
 Figure 4 An input that activates 0th filter in 1st CNN layer.

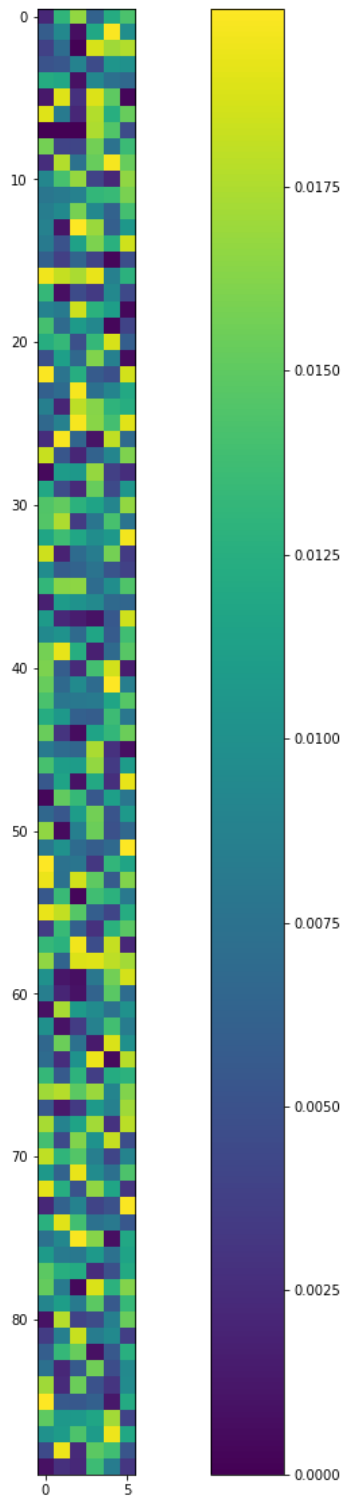


図 5 第 1 層 21 番目の CNN フィルタを活性化させる入力

Figure 5 An input that activates 21st filter in 1st CNN layer.

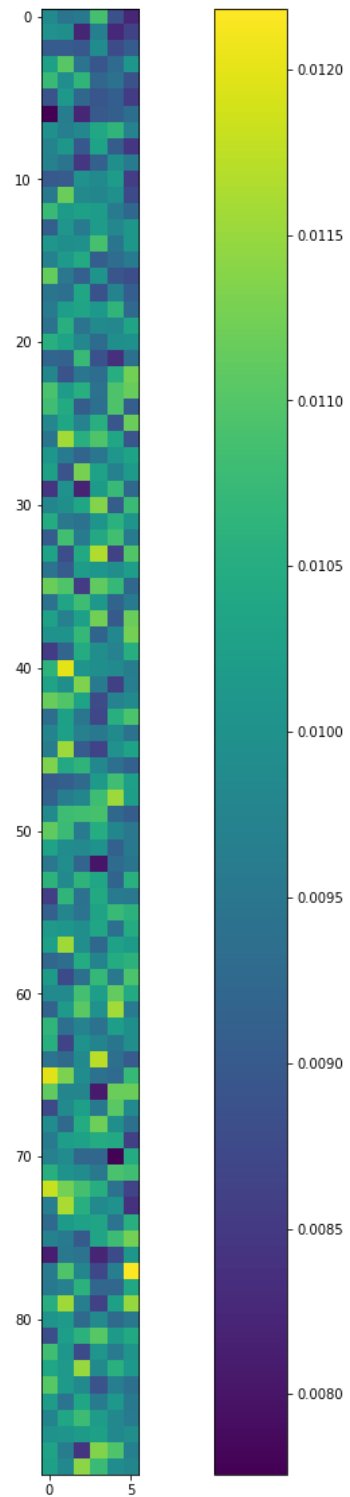


図 6 0 番目から 63 番目まですべての第 1 層 CNN フィルタを活性化させる入力の相加平均

Figure 6 The average of inputs that respectively activates 0th to 63rd filter in 1st CNN layer.

6. おわりに

本研究では、プログラミングコードの編集過程におけるアルゴリズム思考を養うためのツールである「ジグソー・コード」における操作パターンの特徴抽出を試みた。操作パターンの特徴を抽出することにより、学習者に共通の思考過程を理解し、効果的な教材を作成することを狙とした。深層学習を用いて編集ログデータを解析し、学習者に共通の操作パターンを抽出し、操作中の思考過程を考察した。分析手法として、深層学習による操作ログと解答の紐づけを行い、学習したアーキテクチャの特定のフィルタを活性化させる入力を得、その入力を読むことにより操作を行う上でクリティカルなパターンを求めた。結果、「序盤はパズルの盤面をいじることで感触をつかみ」「中盤・終盤は黙考してパズルの盤面を調整している」というパターンを読み取ることができた。

今後、異なる切り口や統計手法での新たな編集の特徴抽出が期待できる。また、意図としては解析の粒度を細かくして正確な特徴を抽出するため、手法としては深層学習の構成とデータ構造に関連して、以下の改善点が挙げられる。機械学習の手法として今回は、操作する行のプログラムの命令文を考慮せずただ機械的にIDを振り分けただけだが、操作する行の内容も含めた解析をすることで、即座に固定される行と試行錯誤を繰り返す行など、命令文の意味も含めた解析が期待できる。また、今回は最終的に得られた行の配列を出力としてデータを作成したが、正解・不正解のパターン分け、あるいは正解からのある尺度を用いた距離、解答に至るまでに要した時間など、どのくらい正解に近いかわるか、どのくらいの早さでどのタイミングで進捗が見られるか、といった尺度を用いた出力にすることで、個々人の学習の姿勢も含めた詳細な解析が期待できる。以上の意図に従い、より適切な機械学習アーキテクチャの構成やデータ構造も探ることにより、より実態に近い学習者の思考過程の考察が行える。

以上の改善点はあるものの、本稿で述べた深層学習による解析を活用することで、プログラミング学習のさらなる定量評価が期待できる。

参考文献

- [1] P.F. ドラッカー(著), 上田惇生(訳). テクノロジストの条件. ダイヤモンド社, 2005, pp.55-69.
- [2] 日本経済再生本部. 未来投資戦略 2018 - 「society 5.0」 「データ駆動型社会」 への変革-
https://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/miraitousi2018_zentai.pdf, (参照 2020-02-16).
- [3] 文部科学省. 小学校学習指導要領 2017.
https://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afieldfile/2019/03/18/1387017_001.pdf, (参照 2020-02-16).
- [4] 松本慎平, 石井元規, 林雄介, 平嶋宗. 学習者による思考及び操作を中心としたプログラミング学習支援システムの設計と

- 開発. 人工知能学会全国大会論文集 第 31 回全国大会, 2017, pp. 1F33-1F33.
- [5] Parsons, D., Haden, P. Parson's programming puzzles: a fun and effective learning tool for first programming courses. *Proceedings of the 8th Australasian Conference on Computing Education-Volume 52*, 2006, pp. 157-163.
 - [6] 森永笑子, 松本慎平, 林雄介, 平嶋宗. カード操作方式によるプログラミング学習システムのフィードバック機能の実装及び評価. 第 43 回教育システム情報学会全国大会, 2018, P2-02.
 - [7] 森永笑子, 松本慎平, 村上瑠香, 林雄介, 平嶋宗. カード操作方式によるプログラミング学習支援システムでの学習過程の可視化方法の提案. *SIG-ALST*, 5(03), 2019, 92-97.
 - [8] Ithantola, P., Karavirta, V. Two-dimensional parson's puzzles: The concept, tools, and first observations. *Journal of Information Technology Education*, 10(2), 2011, 119-132.
 - [9] 石井元規, 松本慎平, 林雄介, 平嶋宗. カード方式を用いたプログラミング学習支援システムの認知負荷の観点による評価. *SIG-ALST*, 5(03), 2017, 11-16.
 - [10] Kumar, A. N. Representing and Evaluating Strategies for Solving Parsons Puzzles. In *International Conference on Intelligent Tutoring Systems*, 2019, pp. 193-203.
 - [11] 川北紘正, 大場みち子, 山口琢. プログラミング思考過程に基づくプログラミング時の行動分析と傾向. 第 81 回全国大会講演論文集, 2019, 521-522.
 - [12] 中村陽太, 大場みち子, 山口琢, 伊藤恵. 学習進度に対応するパズルを利用したプログラミング思考過程の分析. *研究報告コンピュータと教育(CE)*, 2019, 1-9.
 - [13] 大場みち子, 山口琢, 川北紘正. パズルを利用したプログラミング思考過程の分析. *情報教育シンポジウム論文集*, 2019, 152-159.