

公立はこだて未来大学 2017 年度 システム情報科学実習
グループ報告書

Future University Hakodate 2017 System Information Science Practice
Group Report

プロジェクト名

AI するディープラーニング

Project Name

AI Love Deep Learning

グループ名

詰将棋自動生成グループ

Group Name

Automatic Tsume Shogi Generation

プロジェクト番号/Project No.

13-B

プロジェクトリーダー/Project Leader

1015031 黒川大智 Daichi Kurokawa

グループリーダー/Group Leader

1015069 稲田将文 Masafumi Inada

グループメンバ/Group Member

1015031 黒川大智 Daichi Kurokawa

1015069 稲田将文 Masafumi Inada

1015145 中野豪大 Takehiro Nakano

1015152 渡邊凜音 Rion Watanabe

1015225 菊地勇然 Takenori Kikuchi

1014132 岡留司真 Kazuma Okadome

指導教員

竹之内高志 寺沢憲吾 香取勇一 片桐恭弘

Advisor

Takashi Takenouchi Kengo Terasawa Yuichi Katori Yasuhiro Katagiri

提出日

2018 年 1 月 29 日

Date of Submission

January 29, 2018

概要

ディープラーニング (深層学習) とは脳のニューラルネットワークを応用した機械学習の手法である。神経細胞にあたるノードを多層に組み合わせることで従来よりも複雑な学習が可能である。その活用範囲は幅広く、画像処理、音声処理、自然言語処理などの技術に応用されている。しかし、ディープラーニングの実用化は最近のことであり、その活用範囲は未知数である。

一方で、私達はディープラーニングを詰将棋に応用できないかと考えた。詰将棋とは将棋から派生したパズルゲームである。駒が配置された将棋の局面から攻め方が玉方に王手をかけ続けて玉将を詰める事が目的である。詰将棋生成プログラムには先行研究として野下 (1996) のランダムに駒を配置した局面の駒を消すなどの操作を加えることで生成する方法や、広瀬ら (1998) の詰め上がりの局面から1手ずつ戻すことで生成する方法が存在している。しかし、これらの詰将棋生成プログラムは単調な問題を生成しており人間が作ったような面白い詰将棋の生成には至っていない。私達は人間が作った詰将棋から「人間らしさ」を学習できれば、面白い詰将棋が生成できるのではと考えた。そこで、私達は人間が作った詰将棋の特徴を学習し面白い詰将棋を生成するシステムを開発することを提案し実現に向けて活動した。

私達は詰将棋生成アルゴリズムにディープラーニングを用いることで面白い詰将棋を生成することを目的としている。その目的を実現するためには人間が作った詰将棋から面白さの特徴をディープラーニングに学習させなければならない。私達は詰め手順に着目した。自然言語処理において単語の意味を学習するために用いられる Word2vec という手法を応用し、手順から駒の並び方の傾向を学習させることを考えた。この Word2vec の学習済みモデルは入力された手と意味合いが類似する手の候補を出力することができる。

生成方法は、初めに従来の詰将棋生成プログラムで詰将棋を生成する。その詰将棋の詰め手順をディープラーニング学習済みモデルに入力し類似する手順の候補を出力させる。そして、入力された詰め手順を出力された候補手に入れ替える。入れ替えられた詰将棋が詰将棋問題として成立しているならば詰将棋の問題として生成する。この方法で生成された詰将棋は従来の方法より評価が高く、手順が複雑になることで難易度が上がった。

キーワード 深層学習, 詰将棋, Word2vec

(※文責: 稲田将文)

Abstract

Deep learning is the technique of machine learning applying the brain's neural network. By combining nodes corresponding to nerve cells in multiple layers, it is possible to learn more complicated than before. Its range of application is wide, it is applied to technologies image processing, speech processing, natural language processing, and so on. But, implementation of deep learning is recent and its application range is unknown.

On the other hand, we think that a deep learning can apply to Tsume Shogi. Tsume Shogi is a puzzle game derived from Shogi. The purpose is to continue to checkmate the Gyokusho in the state of Shogi. There are two previous researches in field of Tsume Shogi generator. One is the way to put pieces randomly and adding operate such as erase pieces or move pieces to generate Tsume Shogi problem made by Noshita(1996). Other one is the way to remove from final state of Tsume Shogi to generate Tsume Shogi problem made by Hirose et al.(1998). But, these two Tsume Shogi generator could not generate interesting Tsume Shogi problem like human made because the generator made only simple problem. So, we think that if deep learning can learn "Humanity", it can generate interesting Tsume Shogi problem. Therefore, we propose to develop a system to learn the characteristics of Tsume Shogi made by humans and generate interesting Tsume Shogi and worked towards realizing it.

We aim to generate interesting Tsume Shogi by using deep learning for Tsume Shogi generation algorithm. In order to realize that purpose, we must make learning the characteristics of interest from deep learning from Tsume Shogi made by humans. We focused on the procedure to solve Tsume Shogi, we applied Word2vec, which is used to learn the meaning of words in natural language processing, and thought that it learn the trend of arranging pieces from the procedure. This learned model of Word2vec can output candidates of move that are similar in meaning to the input move.

As for the generation method, first, Tsume Shogi is generated by the conventional Tsume Shogi generation program. Procedure for filling the Tsume Shogi is input to deep learning learned model and candidates of similar procedures are outputted. Then, the input procedure is replaced with the output candidate move. If the replaced Tsume Shogi is established as a Tsume Shogi problem, it will be generated as a problem of Tsume Shogi. The Tsume Shogi generated by this method has higher evaluation than the conventional method, and the difficulty level has increased due to the complicated procedure.

Keyword Deep Learning, Tume Shogi, Word2vec

(※文責: 稲田将文)

目次

第 1 章	はじめに	1
1.1	背景	1
1.1.1	詰将棋とは	1
1.1.2	詰将棋の特殊なルール	2
1.1.3	特殊な詰将棋	3
1.1.4	先行研究とその問題点	3
1.2	目的	4
第 2 章	プロジェクト学習の概要	6
2.1	課題設定	6
2.2	前期目標	6
2.3	後期目標	6
第 3 章	課題解決のプロセスの詳細	8
3.1	技術, 知識習得	8
3.1.1	Python とディープラーニングの学習	8
3.1.2	Chainer の学習	8
3.2	どのように詰将棋から特徴を学習させるか (アイデア) の提案と検討	8
3.2.1	画像処理	9
3.2.2	自然言語処理	9
3.2.3	音声処理	9
3.2.4	DCGAN	10
3.2.5	LSTM	11
3.3	中間発表	12
3.4	開発	12
3.4.1	ランダム法の詰将棋生成プログラムの開発	12
3.4.2	使用した将棋ソフト	13
3.4.3	評価プログラムの開発	14
3.5	課題と到達目標の修正	14
3.6	最終発表	15
第 4 章	プロジェクト内のインターワーキング	16
第 5 章	実験	19
5.1	検討した学習手法について	19
5.2	開発したプログラムの動作	20
5.3	データセット	20
5.4	評価について	21
5.4.1	枚数について	21

5.4.2	重みについて	22
5.4.3	玉の開放度	22
5.4.4	配置の広さ	23
5.4.5	駒取りの回数について	23
5.4.6	使用率について	24
5.4.7	玉の位置	24
5.4.8	手数に対する持ち駒の数について	25
第 6 章	結果	27
6.1	学習の結果	27
6.2	詰将棋生成の結果	28
第 7 章	考察	30
第 8 章	今後の展望	31
第 9 章	まとめ	32
	参考文献	33

第 1 章 はじめに

1.1 背景

1.1.1 詰将棋とは

昨今、コンピュータによる本将棋への挑戦は佳境を迎え、ついには人間を超えたと言われるようになった。また、コンピュータにパズルやクイズのようなものを解かせる、作らせるといった試みも数多く存在する。そこで、私たちは将棋かつパズルであり、歴史の長さから数多くのデータがあると考えられる詰将棋も、コンピュータで学習することができるのではないかと考えた。とはいえ、詰将棋と一口に言っても、実に様々な種類の詰将棋があり、将棋とは異なる独自のルールも存在する。下記とは異なるものも詰将棋と呼ばれることがあるかもしれないが、本研究における目的としての詰将棋の定義はここに示されるものとする。

- 詰将棋の概要

詰将棋とは本将棋をもとに作られたパズルゲームである。図 1.1 に示したものが本将棋の初期配置、図 1.2 に示したものが詰将棋における初期配置の一例である。本将棋は二人のプレイヤーがおり、先手後手に分かれ試合を進めるが、詰将棋は一人で行う。本来詰将棋で後手となる手番では、手順を増やす合い駒以外は試合ができるだけ長く続けることができるように動く。詰将棋の挑戦者は先手である攻め方となり、玉方の玉を詰ますことを目指す。駒の動きや詰みの条件などほとんどのルールは本将棋と同じだが、本将棋と異なるルールとして 1.1.2 詰将棋の特殊なルールの項に述べるものがある。



図 1.1 本将棋

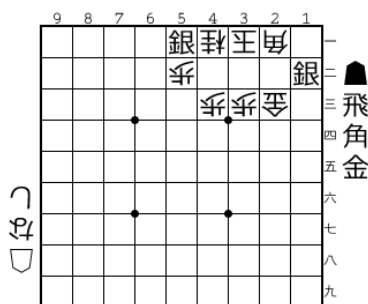


図 1.2 詰将棋

1.1.2 詰将棋の特殊なルール

- 攻め方は毎回必ず王手をかけ続けなくてはならない。
- 持ち駒がある状態で詰ませないまた、ルールとされているわけではないが攻め方は王将を持たないのが一般的である。特別に、攻め方が王将を持つものを特別に双玉詰将棋と呼ぶこともある。
- 問題に余詰めは存在しない。余詰めとは作成時に想定されている本来の詰め手順と異なるにもかかわらず詰んでしまう手順のことである。但し、最後の一手に違う手があっても、それを余詰めとはしない。例として、1.3 図のようにあと一手で詰む、となったとき図 1.4 のように金を玉の真下に打っても、図 1.5 のように金を玉の左下に打っても成立する場合は、これを余詰めとはしない。

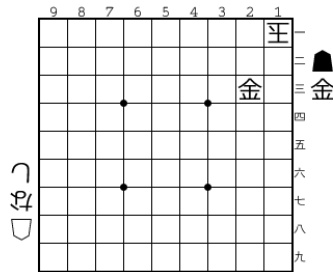


図 1.3 詰み一手前

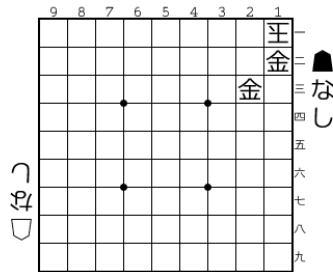


図 1.4 非余詰め例 1

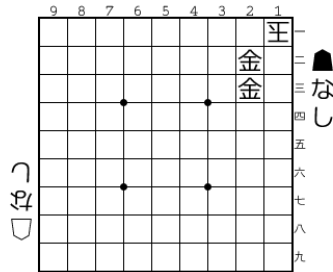


図 1.5 非余詰め例 2

1.1.3 特殊な詰将棋

詰将棋には、上で述べたように独自の呼び名を持つタイプの詰将棋がある。そのうちいくつかを以下に紹介する。

- 双玉詰将棋

前述の通り、攻め方玉方共に王将(玉将)を持つ詰将棋である。この将棋の特徴としては攻め方にも玉将があるので、一般的な詰将棋と違い、玉に対しての相手の大駒の効きをふさぐことを意識しなければならないなど、より実践に近い形の詰め将棋を解くことができる。

- 実践詰将棋

こちらは、呼び方、内容ともに明確には定義されていないが、新聞などに掲載されることも多い、比較的なじみのある形式である。この形式の特徴としては1一香や2一桂のような初期配置の駒が置かれていることや盤面の四分の一程度の範囲に駒が収まっていることがあげられる。双玉の実践詰将棋問題もある。

- 曲詰め

最初の配置や詰上がりの形に特徴がある詰将棋の形式である。最初の配置で文字や図形を作るものを形象型、詰上がりで作るものをあぶり出し、配置と詰上がり共に意味があるものは立体曲詰と呼ぶ。祝い事の際に送る祝賀詰といったものもある。

- 煙詰め

攻め方の玉を除きほぼすべての駒が配置され、詰上がりの際には三枚もしくは四枚の駒で詰める詰将棋のことである。飛車角を除いた35枚で作る小駒煙や歩以外の駒で作る歩なし煙などのバリエーションが存在する。

このように、様々な詰将棋があるが、すべての方式の詰将棋に対応した自動生成プログラムを作るのは困難である。そこで、本研究では特殊でない詰将棋及び曲詰めについて調査を行い、目標を定めた。

(※文責: 中野豪大)

1.1.4 先行研究とその問題点

詰将棋の自動生成に関する研究は二十年以上前から行われている。以下にそれらの先行研究のうち二つを紹介する。野下(1996)[1]は、詰将棋の駒をランダムに配置したのち、一定のルールに従って余分なものとなった駒を消すなどの操作を加えることで、詰将棋を生成する方法(以下、ランダム法)を発表した。広瀬ら(1998)[2]は、詰将棋ではなく本将棋の詰め上がりの局面から1手ずつ戻すことで詰将棋を作成する方法(以下、逆算法)を発表した。これらのアルゴリズムには以下に述べる問題点がある。はじめに、ランダム法の問題点について述べる。ランダム法は、ランダムに駒を配置した後、指定したルールに従って駒を移動する、あるいは削除する事により問題を生成する手法である。このアルゴリズムの問題点は、飛車や角と言った効きが広い大駒が削除する際のルールにかかりやすく、歩のような駒よりも削除されやすい点である。その結果、この方法を用いて作った問題は駒が少なくなる傾向があり、実際に解く時に考えうる局面の数が減ってしまうことから、頭を使う良問ができ辛いという問題がある。次に、逆算法[1]についての問題点について述べる。この方法の問題点は、コンピュータのリソースがボトルネックとなってしまうことで

ある。将棋において一手遡ろうとした時、遡ることが可能な手は一つのみではない。駒を移動させた場合、あるいは何かの駒が打ち込まれた場合、移動させたと考えるなら駒をとった場合、取らなかった場合、とこのように実に多くの局面が想定され得る。こういったものをそれぞれの局面ごとに逆算していくと膨大な数になってしまうことから、計算量も膨れ上がってしまう。この問題の解決方法は単純で、コンピュータの性能を上げる、もしくは手数を減らすことである。しかし、コンピュータの性能はすぐにかできるものではない。また、手数が少ない詰将棋を作った場合、その問題は考えられる局面の数は少なくなり、その結果挑戦者が騙されるような偽物の筋も減ってしまう。このように、従来の生成法では、面白みに欠ける問題が生成されやすいという問題点があった。

ここまで提示したように旧来の詰将棋生成アルゴリズムでは、大駒も手数も少なくなり、面白みに欠けるものができやすいといった問題がある。それぞれのアルゴリズムの問題点をもとに、私たちの詰将棋生成アルゴリズムが満たすべき点として、大駒を削り落とすようなアルゴリズムにならないようにすること、手数の多い詰将棋ができやすいようにすること、この二つを満たすべき点だと考えた。それらを実現するための方法として、ディープラーニングを用いて作ることにした。一つ目の問題点である大駒が削れやすいことに対しては、ディープラーニングの学習対象を大駒が採用されている問題とすることで、大駒が採用された問題ができやすくなるようにする。二つ目の問題点である手数の多い詰将棋が出来辛いことに対しては手数の多い問題から学習することで解決を図る。どちらの改善案も、詰将棋のデータベースから適切に学習するプログラムを考えることが必要となる。当グループではいくつかの手法で実装を試み、Word2vecを用いる手法を採用することとした。

(※文責: 中野豪大)

1.2 目的

このグループで行ったのは、上記の通り、詰将棋の自動生成である。しかし、既存の研究で行われている生成方法において、例えば、ランダム法では、大駒が失われやすいことや、逆算法では、用いるコンピュータの性能にもよるが、長手数の詰将棋を生成できないなど、単調でつまらない詰将棋を生成してしまいやすいという欠点があった。

その問題を解決するために私たちが注目したのは、プロジェクト名に含まれているディープラーニングという技術が特徴抽出に秀でていているということである。そのディープラーニングを用いて、人間が作った詰将棋の特徴をうまく抽出できないかということを考えて。その場合、ディープラーニングをどのように使うかの候補として、画像処理（畳み込みニューラルネットワークなど）や自然言語処理（リカレントニューラルネットワークや LSTM など）といった手法が挙げられる。

画像処理の手法なども検討したが、最終的には、その中から、私たちは、自然言語処理の手法を用いることとした。具体的には、Word2vec という手法である。この Word2vec とは、端的に言えば、自然言語処理において、単語の分散表現を求めるためのもの [3] である。言い換えると、その単語の意味を、低次元の密な実数値ベクトルで表現したものである。また、入力は一コーパス、出力は各単語の分散表現を用いる。私たちは、この分散表現を求めるという部分に着目し、詰め手順における、その手番や駒の種類などの部分を入力にすることで、その手順の配列から、詰将棋における駒の意味を学習させることができるのではないかと考えた。

また、今回のプロジェクトでは、詰将棋を生成することが必要なため、ランダム法の詰将棋を生

AI Love Deep Learning

成し, Word2vec における候補手と元の手の入れ替えを行い, 将棋ソフトに通すことで新しい詰将棋を生成する.

さらに, Word2vec による変更により作られた詰将棋と変更を行うための詰将棋のどちらが良いかを評価するための関数が必要になってくる. 今回は, 広瀬ら (1998) の論文で用いられた評価の関数を自分たちで適宜変更を行いながら, より良い関数を作成する.

これらから, 私たちは, Word2vec により人間が作った詰将棋の特徴を学習し, より面白い詰将棋を生成することを目的とする.

(※文責: 菊地勇然)

第 2 章 プロジェクト学習の概要

2.1 課題設定

本プロジェクトグループの目的を達成するために、大きく分けて 4 つの課題を設定した。

1 つめは、知識と技術に関するものである。まず、本プロジェクトのメインテーマである、ディープラーニングについて学ぶ必要があると感じ、ディープラーニングの基礎的な知識を学ぶことを課題とした。これには、書籍と Web 上の情報を使うこととし、参考書として、「ゼロからつくる Deep Learning[4]」を購入した。ディープラーニングについて学んでいくうちに、目標達成のために必要なものとして、数あるディープラーニングの手法から適切なものを選ぶこと、学習を行うための適切なコンピュータが必要であることが分かった。次に、Python を学び、ディープラーニングを用いたプログラムを作成できるようにすることを課題とした。機械学習を扱うフレームワークを使えるようにして、開発に取り掛かれるようにした。今回は、Chainer というフレームワークを選択し、これを学ぶことを課題とした。また、詰将棋に関する知識が足りないと感じ、詰将棋に関する基礎的な知識を学ぶことも課題とした。

2 つめは、開発環境についてである。機械学習を行うために、GPU を搭載したコンピュータを用意することとした。また、Python や Chainer についての学習には、Anaconda を各自のノートパソコンにインストールし、Jupyter Notebook を使用した。その後、実際の開発には向かないと感じ、IDE である、PyCharm を使って開発を進めた。

3 つめは、作成するプログラムについてである。我々は、2 つのプログラムを作成することにした。ランダム法を用いて、詰将棋を生成するプログラムと、Word2vec の手法を用いて、棋譜を学習するプログラムである。これらのプログラムを作成することを課題とした。また、詰将棋を客観的に評価するための評価基準を確立することも課題とした。

4 つめは、グループワークについてのものである。1 年を通してのスケジュールを決めた後、グループ内の進捗を管理するために、マイルストーンを設けた。また、毎回の活動で議事録をつけることとした。

(※文責: 渡邊凜音)

2.2 前期目標

当初の目標は、前期のうちに、3 手詰の詰将棋を自動で生成することであった。しかし、進捗状況から、詰将棋生成のアイデアと手法を提案することを、到達目標として掲げた。

(※文責: 渡邊凜音)

2.3 後期目標

後期では、ランダム法を用いた盤面生成プログラムを作成すること、Word2vec を用いた、駒の入れ替えを行うこと、評価値を求め、比較することを最終目標として掲げた。

第 3 章 課題解決のプロセスの詳細

3.1 技術, 知識習得

Python の学習には「ゼロから始めるディープラーニング」という書籍を用いてディープラーニングとも絡めて学習していった。またフレームワークに Chainer を使って実装することにしたため、「はじめての深層学習プログラミング [5]」という書籍を使い、Chainer についても学習した。

(※文責: 岡留司真)

3.1.1 Python とディープラーニングの学習

私達はディープラーニングをプログラムするには使用する言語を Python にすることが適切だと考えた。なぜなら、Python にはディープラーニングをプログラムするためのフレームワークがあるからである。Python を使用するための学習と並行してディープラーニングの知識習得もしなくてはならなかったので、「ゼロから始めるディープラーニング」を用いて学習した。この書籍はディープラーニングを Python でプログラミングしていく内容となっている。しかし全てを読み、理解するには時間が多くかかってしまうため 4 章まで進めることを定めて、学習に取り組んだ。ディープラーニングは同じ書籍を用いての学習であった。ディープラーニングの基本的なモデルや、使用する関数を学習した。

(※文責: 稲田将文)

3.1.2 Chainer の学習

Chainer についての学習には、参考書と同時に、Chainer のホームページを参考にした。まず、Chainer を扱える環境として、Python のバージョンを 3.6.0、Chainer のバージョンを 2.0.0 で構築した。また、ツールとして、Jupyter Notebook を使用した。これにより、プログラムの実行結果を段階的に保存することができ、プログラムの動きを理解することが容易になった。学習には、Chainer のサンプルプログラムを用いた。そして、プログラミングに必要な要素として、計算モデルの定め方を学んだ。

(※文責: 渡邊凜音)

3.2 どのように詰将棋から特徴を学習させるか (アイデア) の提案と検討

課題解決のために、既存の機械学習の手法の中から、詰将棋に応用できる手法がないかを、画像処理、自然言語処理、音声処理の 3 つのグループに分かれ、それぞれの観点から、適用できそうな

手法を調べた。その結果、画像処理の点からは、DCGAN という手法が、自然言語処理の点からは、LSTM という手法が詰将棋に適用できると考えた。この節では、3つのグループの調査結果と DCGAN, LSTM について述べる。

(※文責: 渡邊凜音)

3.2.1 画像処理

画像には、2次元の座標と、モノクロであれば1チャンネル、カラーであれば3チャンネルの色情報がある。これを詰将棋に応用する方法がないか考えた。詰将棋は、2次元の座標と、駒の種類の数だけのチャンネルがあるといえる。このように考えれば、詰将棋の盤面を、画像と同じように表現することができ、同じようにディープラーニングの手法が使えると考えた。画像処理に関する手法はいくつかあるが、今回は画像生成に注目して、DCGAN という手法が適していると考えた。DCGAN の詳細については後述する。

(※文責: 渡邊凜音)

3.2.2 自然言語処理

自然言語処理と、詰将棋には以下の共通点があると考えた。

- 入力に離散的である (文字, 単語, 駒, 場所)
- 並びが一方向である (文章, 棋譜)
- 単語列の順序に比較的強い制約 (文法, 詰将棋のルール)

また、自然言語処理を難しくする要因として、単語の数が膨大であることと、入力が可変長であることがある。これは、詰将棋と比較すると、単語の数と駒の数に対応すると考えると、駒の種類は10種類と少なく、入力については、3手詰めや、5手詰めなど種類に応じて固定することができるため、無視できるものであると考えられる。以上のことから、自然言語処理の手法を詰将棋に応用することは可能であると考えた。そこで、我々は、自然言語処理に関する手法である LSTM という手法が適していると考えた。LSTM の詳細については後述する。

(※文責: 渡邊凜音)

3.2.3 音声処理

音声処理の中から、特に、音声認識に注目して手法を調べた。音声認識のメカニズムについて解説する。まず音声データのグラフを細分化し、そこから機械学習を用いて、音素を抽出する。そして、音素と辞書データを照らし合わせて文として出力するものである。これを詰将棋に置き換えると、入力された棋譜を1次元に直して入力するという手法が考えられる。また、音楽生成についても調べた。コード進行を生成するディープラーニングの例から、コードの遷移を、盤面の遷移としてとらえ、詰将棋の生成に応用できないかということも考えた。しかし、音声処理を詰将棋に応用することは難しいと考え、別の方法を使うこととした。

(※文責: 渡邊凜音)

3.2.4 DCGAN

DCGAN という手法を紹介し、その応用方法について述べる。この手法は、画像処理の手法である。このような画像処理の手法が有効であると考えた理由としては、AlphaGo や Ponanza Chainer などが CNN（畳み込みニューラルネットワーク）と呼ばれる手法で、盤面を様々な特徴量として入力し、そこから最善と思われる手を出力させるということを手で行っていたためである。

GAN (Generative Adversarial Network) とは、ディープラーニングを用いた生成モデルの一つで、訓練データから学習し、その学習したものに似たものを生成するモデルの一つである。また、adversarial (敵対的な、敵対関係にある) という単語の通り、generator というネットワークと discriminator というネットワークが存在し、それら二つの間に敵対関係が存在する。敵対関係としては、generator は discriminator に見破られないようにするために、精度の高いものを作るように学習し、逆に、discriminator は generator が生成したものをできるだけ、偽物として判定できるように学習していくというものである。

DCGAN(Deep Convolutional Generative Adversarial Networks) では、その名の通り、GAN に、画像処理分野で大きな成功を得ている畳み込みニューラルネットワークを用いている。それにより、より精度の高い画像を生成できるようになるというものである。generator には逆畳み込み層を用いることで画像を生成し、discriminator では畳み込み層により、画像が目的のものと合致しているかを判別することができる。

そもそも、ディープラーニングが画像処理を得意としている理由として、各ピクセルを RGB の値として配列化することができ、数値として扱うことができるからであると考えられる。

詰将棋生成に話を戻す。詰将棋の特徴として、

- 初めから駒の配置は決まっている
- 駒の種類が限られている
- 全く同じ配置は 1 つもない

というものがある。

上で述べた DCGAN と詰将棋の特徴を考慮して、私たちは次のように、詰将棋に応用することを考えた。そのイメージが図 3.1 である。

- 盤面を 9×9 × 特徴量の多次元配列として捉える
- 特徴量には駒の種類などが該当する
- 問題にある程度の共通性を持たせるため、学習の際に扱うのは、配置に特徴のある詰将棋 (曲詰め)
- 生成するのは詰め上がり局面とし、逆算法の詰将棋生成法を用いて、詰将棋を完成させる
- 柿木将棋等の将棋ソフトを用い、詰将棋として成り立っているかを確認する

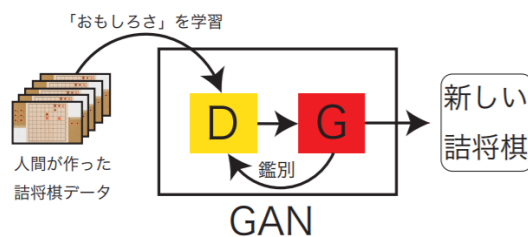


図 3.1 DCGAN を用いた詰将棋生成のイメージ

しかし、構想が固まってくるとともに以下のような課題点や疑問点が見つかった。例としては、

- 持ち駒などの 9×9 の盤面に入っていない情報をどう入力するか
- 駒の移動できる位置や利き筋等をどのように表現するか

といったものである。

これらの議論をしていたが、11月に入った段階で、Word2vecを用いた手法へと一本化することとなったため、この手法は凍結状態となった。Word2vecの手法を優先した理由としては、

- 詰将棋生成プログラムに関して、ランダム法と逆算法の二つを作らなければいけなくなり、時間と人員が足りなくなるから
- 配置に特徴のある詰将棋を用いようとしていたが、データの絶対数が足りないことから

というものがあつた。

(※文責: 菊地勇然)

3.2.5 LSTM

LSTM(Long Short-Term Memory)とは1997年にH. Hochreiterらによって提唱されたRNN(Recurrent Neural Network)を拡張した学習手法である。まず、元となるRNNについて説明する。RNNとは時系列データを扱うことのできるニューラルネットワークである。時系列データとはなにかというと、例としては文章があげられる。文章では直前の単語のみではなく、文全体でようやく意味が理解できる。「私は黄色いペンを3本持っている。」という文章があつたとき、「私は黄色いペンを3本」というところまで見れたなら「持っている」を推測することができるかもしれないが、直前の「ペン」という単語だけを見て「持っている」を推測するのは困難である。このように、直前だけでなくさらに前の情報も保持する必要があるデータを扱うためRNNの手法が開発された。しかし、RNNの手法では短い時系列データは扱うことができたが、長期の依存関係を学習すると勾配消滅もしくは勾配爆発という現象が起こることから学習が難しかった、そこで、RNNに忘却ゲート、入力ゲート、出力ゲートというものを追加することにより長期の依存関係を学習することを可能にしたものがLSTMである。

(※文責: 中野豪大)

3.3 中間発表

スライド作成は冒頭の共通部分を除き、A グループ、B グループでそれぞれ個別に作成し、最終的にそれらを統合する方針で決定した。

冒頭の共通部分には各グループに共通するプロジェクト発足の背景と各グループが共通して行った活動について記述した。前者にはディープラーニングについての大まかな説明や、このプロジェクトの目的について記述した。後者にはグループが分かれる前に行ったニューラルネットワークや Python についての基礎的な学習や、各メンバが考えてきたテーマから二つのグループに分かれるまでの過程を記述した。また、B グループのスライドについてはまず詰将棋がわからない人がいる可能性があるため詰将棋の大まかなルールなどの解説や詰将棋作成の難しさを説明した。次に詰将棋の自動生成のアルゴリズムなどの詰将棋自動生成の歴史について解説し、本グループが取り組むテーマでの背景を説明した。以上のプロジェクトの背景を説明したうえで本グループの最終的な目的について説明し、それを実現するために本グループが提案したアイデアについての説明を行うという流れで中間発表のスライドを発表した。表 3.1 は評価シートの内容を表す。また中間発表の発

評価	発表技術	発表内容
5 点	0	1
6 点	4	3
7 点	7	14
8 点	13	6
9 点	0	1
10 点	2	1

表 3.1 中間発表の評価シートの点数

表技術の評価点の平均は約 7.58 点、発表内容は約 7.23 点となった。発表技術に対して発表内容が低めな理由としては二つあり、一つは中間発表の時はまだ実装するための手法が大雑把で具体的ではないものだったということ、そのため評価シートのコメントにもその辺りを指摘するコメントが多く見受けられた。またもう一つの理由として、面白い詰将棋の定義について説明が不足していたということ。面白さの定義に疑問を感じている評価シートが多く見受けられたため、そういった内容を説明する際に入った内容へのぼかしが、発表内容の評価が低下した一因であったと考えられる。

(※文責: 岡留司真)

3.4 開発

3.4.1 ランダム法の詰将棋生成プログラムの開発

このプログラムは元々、前述の LSTM を用いた生成の班が制作を開始したものである。このプログラムに LSTM の学習で得られたパラメータを用い、詰将棋の特徴をフィードバックすることで人間が作った詰将棋の特徴を踏襲した、より面白い詰将棋の生成を想定していた。制作に関しては、先行事例で挙げた通り、野下 (1996) が論文で紹介していたものを参考にした。元々、逆算法の

プログラムの製作も考えていたが、余詰等の対処に追われ、着手することはできなかった。8月からの作成となり、細かい修正を加えながらの作成となった。その流れを時系列順に説明していく。

作成を始めた8月、この段階では駒を数字として扱い、ランダムな配置で盤面を作るという部分の製作が完了していた。

9月には、チームの再編があり、AI班とランダム生成班に分かれた。この段階では、ランダムな配置の盤面を詰将棋を記録する際のデファクトスタンダードとも言える kif ファイルに出力した。そして、将棋ソフトである柿木将棋を用いて、生成した詰将棋が詰将棋として成立しているか、また、余詰がどの程度あるかがわかる状態となっていた。しかし、予想以上に余詰が多い状態であったため、対策を講じるための話し合いを行った。その話し合いの結果、

- 駒の移動履歴を記録する関数
- 王手をかけている駒を判別するための関数

を作成することになった。作成した結果、以前よりは余詰が減少したため、有効な手段であったとし、採用することとなった。

10月には、AI班が制作するディープラーニングを用いたAIとの連携をするためのジョイント部を作る活動を行った。これは、AIが学習したパラメータを詰将棋に生かし、元の生成プログラムのランダムな部分に偏りを持たせることが目的である。話し合いを行う中で、MCMC法（マルコフ連鎖モンテカルロ法）という手法が候補として挙がった。しかし、MCMC法に関する知識が足りなかったため、専門書を読むなどの学習を重ねた。

結局、11月にグループ全体の話し合いで、Word2vecを使った手法に切り替えたため、MCMC法は用いなかった。そこで、この詰将棋生成プログラムに関しては、修正などのブラッシュアップをメインに活動をシフトチェンジした。この間に、

- 持ち駒のランダム性の向上
- 詰将棋自体のルールに則っていなかった部分の修正
- バグの除去など

を行った。

しかし、最終的に、完全にランダムな状態から生成を行ったため、一つの詰将棋生成に時間がかかるという問題が残った。

(※文責: 菊地勇然)

3.4.2 使用した将棋ソフト

詰将棋を生成するプログラムを作成するにあたって将棋ソフトを使用した。なぜなら、将棋ソフトを使用することにより、詰将棋の詰め手順や余詰などが正確に知ることができるであろうと考えたからである。使用した将棋ソフトは「柿木将棋 IX」である。開発者である柿木様に使用の許可を頂き、本プロジェクトで使用している。

(※文責: 稲田将文)

3.4.3 評価プログラムの開発

詰将棋の自動生成後、生成された問題の出来を評価しより良い詰将棋ができたかを定量的に評価することを目的とし、評価プログラムを開発した。11月上旬にプログラムの設計、必要項目の洗い出しを行った。必要項目を設定した後、実際にプログラムで生成した詰将棋とデータベースの詰将棋の問題の粗点をだし、適切な係数を導くべく試行を繰り返した。当初のプログラムでは評価プログラムにおける評価項目は枚数、重み、玉の開放度、駒取り、配置の広さ、玉の位置、使用率の7つの予定であったが、試行していくうちに評価項目に不足が感じられるようになった。そこで、手数に対する持ち駒の数を追加し、それぞれの項目に適切な係数の設定を行い、プログラムを完成させた。

(※文責: 中野豪大)

3.5 課題と到達目標の修正

後期活動の中盤に、当初設定した到達目標を達成することが不可能な見込みとなったため、目標の修正を行った。

はじめに当初設定した到達目標について再度確認する。AIによる詰将棋生成を行うにあたり、本グループでは図3.2のようなシステムを開発することを考えていた。

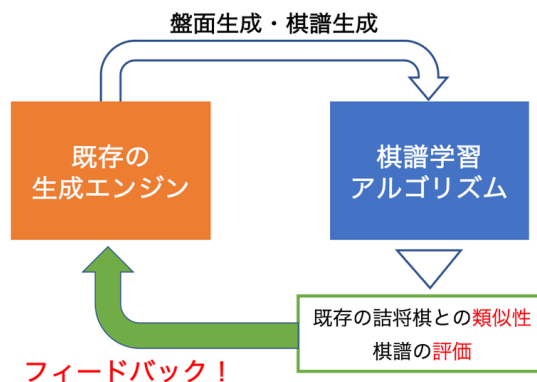


図 3.2 提案された詰将棋生成システム

このシステムは (1) 既存のアルゴリズム「ランダム法」により詰将棋問題を生成する (2) 人間の作った詰将棋問題から学習を行った AI が、生成した詰将棋問題を評価する (3) ランダム法に評価をフィードバックする (4) 以上を繰り返す、という 4 ステップにより「面白み」のある詰将棋問題を生成するものである。

しかし後期活動の中盤の時点で、上記における (2)・(3)・(4) は未完成の状態であった。また、(2)・(3) の開発には解決の難しい課題があった。それは、ランダム法における初期配置の誘導が難しいという問題である。ランダム法では、盤面にランダムに駒を配置した状態から詰将棋を生成する。本グループが当初提案していたアイデアは、(2) のステップで詰将棋を評価し、その結果を (3) でランダム法へフィードバックする、というものであった。フィードバック結果は次に生成される詰将棋に反映される必要があるが、そのためにはランダム法が最初に設定する初期配置を、評価が高まる方向へ誘導する必要がある。しかし、ランダム法へフィードバックされるのは「評価値」という単一の指標のみであり、「どのような詰将棋の評価が高いのか」についてはフィードバッ

クされないため、初期配置を誘導することができないという問題があった。

この欠陥を解決するためには提案していたシステムの構造を再検討しなくてはならないが、欠陥が見つかったのは後期活動中のことであり、再検討する時間は残されていないと考えられた。このことから、我々は当初設定した到達目標を下方修正することとした。結果、今年度の活動では実用的な詰将棋生成システムを開発するのではなく、AIによる詰将棋生成アルゴリズムが従来のアルゴリズムより「面白い」詰将棋の生成に有効であるかを、実験によりに確かめることを目標とすることにした

(※文責: 黒川大智)

3.6 最終発表

最終発表では、主に詰将棋の学習に用いた手法と実験の概要・結果について発表を行った。はじめに本グループがこの問題に取り組んだ背景について説明を行い、続いて詰将棋の学習を行うアイデアについて具体的に示した。次にアイデアが「面白い」詰将棋の生成に有効であるかを確かめる実験の流れを、フローチャートをもちいて説明し、その結果がどのようになったのかを示した。発表後の評価は表 3.2 のようになった。平均評価点は「発表技術」で 7.6 点、「発表内容」で 8.0 点で

評価点	発表技術	発表内容
4 点	2	2
5 点	6	2
6 点	4	3
7 点	10	10
8 点	14	16
9 点	7	6
10 点	6	10

表 3.2 最終発表の評価シートの点数

あった。中間発表と比較すると「発表技術」で 0.07 点、「発表内容」で 0.7 点高評価であった。

中間発表時に比べて評価が上がった一方で、評価シートのコメント欄を見ると「専門用語が多くわかりにくい部分があった」、「ディープラーニングの仕組みの部分について詳しく説明してほしい」という意見が散見された。この理由としては、実験に用いた手法それ自体よりも、「その手法をどのように詰将棋に適用したのか」というアイデアや「実験全体の流れ」を優先して説明したためであると考えられる。反対に、学生よりも専門用語を理解していると思われる教員や一般の方からの評価は高い傾向が見られた。

(※文責: 黒川大智)

第4章 プロジェクト内のインターワーキング

- 黒川大智

プロジェクト開始直後は、テーマ設定や学習などについて指示を行った。具体的にはメンバ全体でのブレインストーミングのファシリテーションや、機械学習の学習本の提案、Python 学習サイトの提案を行った。グループ分離後は、(1)「面白い」詰将棋を生成するためのアイデア出し、(2) 機械学習全般に関わる開発と指揮、(3) 発表時の指揮を行った。(1)については、「どうすれば詰将棋問題を機械学習で生成できるのか」という問に対して、それを解決するアイデアをメンバに提案した。この過程で、参考になるような論文や書籍をメンバに提供した。(2)については、LSTM や Word2vec 等の機械学習手法について調査し、どれが詰将棋の「面白さ」学習に適しているかを検討した。さらに、検討した手法を詰将棋データに適用するための方法を考え、実装した。また、対外連絡係として全日本詰将棋連盟の荒川様へのメールやデータの受け渡しなどを行った。(3)については、発表時に用いるスライドやポスターのデザインを担当した。スライドやポスターは基本的なデザイン（色やページ構成など）を統一し、わかりやすさを重視した。さらに、発表原稿の作成や発表の仕方などについて指揮を行った。また発表当日には、質の高い発表を行うために、発声練習を提案し実行した。

(※文責: 黒川大智)

- 稲田将文

前期は必要な知識を習得しなくてはならなかったため Python とディープラーニングの学習に着手していた。Python を用いてディープラーニングのプログラミングをすることで Python の基本的な使用方法を学習した。また、ディープラーニングに関する知識はノートを書き、まとめることで後の学習や使用したいときに役立つようにしていた。本グループは詰将棋生成をすることが目的である。そのため、それに関する論文や先行研究などを書籍やインターネットを用いて調べた。調べていく中で詰将棋生成の手法が見つかったので、その手法に基づいてプログラムでの実装を試みた。長期休暇である夏休みでは、午前9時からインターネット上で会議を開きプログラムの作成をした。

後期は基本となる詰将棋生成プログラムと余詰め削除のプログラムを作成した。詰将棋生成プログラムは従来方法であるランダム法を参考にし、夏休みを利用して作成した。詰将棋の問題として成り立っているかの判定はできなかったが、柿木将棋を用いて判定するプログラムをメンバーに作ってもらい、ランダム法による生成プログラムが作成できた。余詰め削除のプログラムは、玉将に王手をかけている駒以外を削除するプログラムで、駒の種類ごとにプログラミングした。また、最終報告会では発表とスライドの作成を手伝った。発表で用いられるスライドは整列させることを心掛け、担当の教授方にレビューをもらい作成した。発表はスライドを用いた発表だったため自宅で練習をし、発表の直前までプロジェクトのメンバー全員で練習を重ねた。発表本番では前半と後半のグループに分かれて発表した。私は後半での発表であったため、前半のグループが発表している時間は他のプロジェクトの発表を聞きにいった。発表本番は誰にでも聞こえるような大きな声で発表することを心掛け

た。発表後のアンケート結果に声が大きかった等の書き込みがあったので、目標が達成されていた。2月には秋葉原での発表会に参加する。しっかりとプロジェクトの内容を伝えられるように努めたい。

(※文責: 稲田将文)

- 中野豪大

前期の初めごろはディープラーニングへの理解を深めることを目的に活動した。最初はグループ内ではじめての深層学習プログラミングという共通の教科書を用いることで知識の収集と共通化を同時に進めた。その後、三つのグループに分かれ、活動を行った。三つのグループの作業はそれぞれディープラーニングにおいて画像処理、音声処理、自然言語処理で使われる手法などを調べてくることであった。7月ごろにグループ分けについての話し合いを行い、私の提案した三人グループ制となった。このとき、グループの分け方としては詰将棋の自動生成に GAN を採用するグループと LSTM を採用するグループで分けた。夏休み頃には、詰将棋を生成するために必要なプログラムについて調べ、生成に必要なプログラムなどの設計などについて話し合った。また、グループでランダム法を用いた詰将棋生成プログラムの設計、必要な処理について話し合いを進めた。夏休み終了後には、スケジュール確認や人員の割り振りなどを行い、継続的に開発が進められるように一つ先の項目を学習するものと実装するものに分け、人員配置を考えた。11月からは評価プログラムの作成に向け活動した。最初にどのような評価項目が必要か考案し、菊地君とデータ集計を行ったうえで評価計算式を設定した。

(※文責: 中野豪大)

- 渡邊凜音

はじめに行った、プロジェクトで行う具体的な内容の話し合いで、ゲームに関するテーマを提案した。前期の間は、主にディープラーニングと Python についての基礎的な知識を、「ゼロから作る Deep Learning」や Web 上の情報を使い学習した。機械学習についての学習をグループ内で分担して調べることがあり、私は、音声認識に関連する手法から、応用できるものがないかを調べた。また、活動の議事録を数回つけた。中間発表では、ポスターの作成を担当し、スライドの発表と、ポスターでの発表を2度行った。夏季休業の間は、棋譜のデータを扱いやすいように変換するプログラムを作成した。後期では、アイデア出しの話し合いを多く行った。また、プログラムに関してはアドバイスに徹底した。最終発表では、中間発表同様、ポスターの作成を担当し、スライドの発表とポスターでの発表を2度行った。2月には、秋葉原で行われる発表会に参加する予定である。

(※文責: 渡邊凜音)

- 菊地勇然

まず、前期では3つの視点(画像処理、音声処理、自然言語処理)から詰将棋を捉えてみるという時期があり、その際には、画像処理について担当した。次に、画像処理と自然言語処理のチームに分かれた際には、画像処理班として、DCGANの調査を行い、サンプルプログラムを実行するなどの作業を行い、共有した。また、DCGANは画像に用いる手法であるが、画像と同じように扱うため、詰将棋の盤面をどのように多次元配列化するか調査や

議論を行った。

後期に入り、ランダム法による詰将棋生成プログラムの製作に回った。最初は、自分が全く関わっていなかった部分であったため、プログラムを読むところから始めた。製作中には、プログラムを組む際に、Python の仕様などで詰まった部分を調べ、メンバーに共有した。また、11 月途中から、評価プログラムの開発も行った。

(※文責: 菊地勇然)

- 岡留司真

前期においてはまず、Python やディープラーニングについて学習を行った。大まかな学習が終わりテーマが決まった後は音声処理班と画像処理班、自然言語処理班に分かれ、その際には画像処理班に配属された。そして詰将棋と関係のない音声処理を除外し、残りの二つでグループの再編成を行った際には画像処理班として活動した。中間報告書を書く際に各メンバーがそれぞれ書いたものをまとめて TeX で一つの文書に統合した。また後期でさらに班の再編成を行い、AI 班とランダム生成班に分かれた際には AI 班に配属された。はじめは主に画像処理関係の AI について調べていたが画像処理 AI の実装には不確定要素が多く実装できるかどうかわからなかったため、自然言語処理に一本化した。また Word2vec で実装することに決定した際、学習させるためのデータづくりを担当した。学習の結果をグラフ化したり学習したモデルを用いて、詰将棋プログラム実装班から受け取った棋譜の入れ替えを行い、その棋譜を改善した。

(※文責: 岡留司真)

第 5 章 実験

5.1 検討した学習手法について

本グループでは人間が作った詰将棋の「面白さ」を学習する方法として、以下の 4 つの機械学習手法を検討した。

- (1) DCGAN
- (2) CNN
- (3) LSTM
- (4) Word2vec

このうち、(1) (2) は画像処理を応用した学習手法、(3) (4) は自然言語処理を応用した学習手法である。画像処理を応用した手法では、将棋の盤面を 9×9 の画像として捉えることにより、駒の配置などの情報から詰将棋の「面白さ」を学習させようと考えた。一方、自然言語処理を応用した手法では、詰将棋の「詰め手順」を「文章」として捉えることにより、詰め手それぞれの相互関係から「面白さ」を学習させようと考えた。

次に各学習手法のメリットとデメリットについて示す。(1) (2) の手法を用いるメリットは、いわゆる「曲詰」と呼ばれる詰将棋を生成できる可能性がある点である。「曲詰」とは図 5.1 のように、詰め上がり局面が何らかの意味を持つ図形となるような詰将棋のことである。

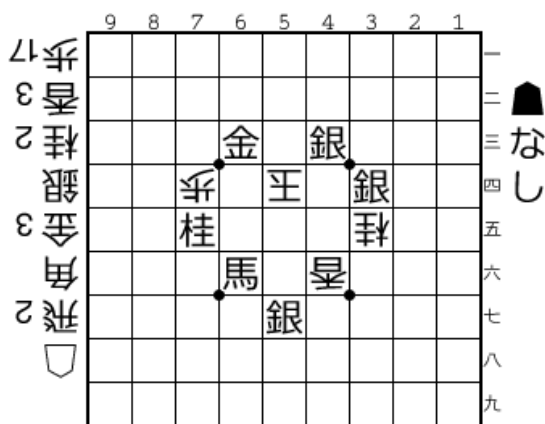


図 5.1 曲詰めの詰め上がり図

「曲詰」の詰将棋は芸術的な側面が強く、詰将棋の「面白さ」の 1 要因になっていると考えられた。一方デメリットとしては、駒自身の情報という、通常の画像にはない情報の学習が難しい点がある。通常の画像はピクセルの場所と色情報 (RGB) により成立しているが、詰将棋は駒の進む方向や駒同士の関係性 (取った・取られるなど) により成立している。従って画像処理手法を用いる場合は、駒の情報をいかに付与するのが課題になると考えられた。(3) (4) の手法を用いるメリットは、自然言語を学習することと何ら変わらない方法で詰将棋を学習させることができる点である。通常、自然言語処理では「文章」とそれを構成する「単語」の集合体である「コーパス」を用いて学習を行う。これを詰将棋の学習に応用する場合、「文章」は詰将棋の「詰め手順」に相当し、「単語」は詰将棋の「詰め手」に対応する。従って、通常自然言語の学習と同じように詰将棋を学習させることができると考えられた。一方デメリットは、盤面の見た目や配置を考慮しない点で

ある。この手法は盤面の初期配置や詰め上がりの情報を考慮しない手法であり、詰め手順における駒の相互関係からしか「面白さ」を学習させることができない。従って先に述べた「曲詰」のような、盤面の配置が面白さに影響するような詰将棋は生成できないと考えられた。

最終的に、本グループではデータセット数や難易度の観点から自然言語処理手法を用いることとなった。次に、(3)と(4)のどちらを用いて詰将棋問題を学習させるかを検討した結果、文章における単語の意味推測に用いられ、詰将棋の学習とより親和性が高いと思われる(4) Word2vecを採用することとなった。

(※文責: 黒川大智)

5.2 開発したプログラムの動作

私たちは Word2vec を利用して人間が作った詰将棋の特徴を学習し面白い詰将棋を生成するシステムを開発しようと考えた。Word2vec とは隠れ層と出力層の 2 層からなる単純なニューラルネットワークで、次々に単語を読み込ませて重みを学習させていく。Word2vec のネットワークの構造として Skip-gram と C-Bow と呼ばれる 2 種類があり、今回は Skip-gram でモデル化した。Skip-gram とはある単語が与えられた時その周辺の単語を予測するためのモデルである。しかし今回は意味の近い単語の出力を目的としていたため、周辺単語の予測はせず単語の分散表現の Cos 類似度から注目した単語と意味の近い単語を出力した。Cos 類似度とは注目した単語ベクトルと他の単語ベクトルの内積から、単語ベクトル同士が似た意味を持っている(単語ベクトル同士のコサインが大きい)かどうかを調べる手法である。またそのモデルが学習した単語の集合をボキャブラリと呼ぶ。本グループではこのボキャブラリを詰将棋で使われる単語の集合とし、詰将棋において攻め方が玉方を詰ませるまでの手順を詰将棋を Word2vec に適用する際の文章とした。そうすることで手順の配列の傾向から詰将棋における駒の意味を学習し、学習したモデルに従ってランダムで生成された詰将棋の詰め手順を変更することで、より「面白い」詰将棋ができるのではないかと考えた。

まず、学習したモデルにランダム生成された詰将棋の詰め手順を適用することで、詰将棋における類義語が出力される。この詰将棋における類義語は、人間が作った詰将棋の中でも詰将棋パラダイス(以下、詰めパラ)と呼ばれる雑誌に掲載された詰将棋であり、人間にとって「面白い」詰将棋から学習され、出力されたものである。そのため、ランダムで生成された詰め手順より「面白い」詰め手順を出力することができる。次に、詰め手順のなか所を変更し、それに合わせて盤面も変更した後、柿木将棋というソフトで変更した詰将棋が詰んでいるかの検査を行った。そして、詰んでいればその詰め手順を Word2vec により改善された詰将棋とし、詰んでいなければ変更前の詰将棋に戻り、変更する箇所を変えた。

(※文責: 岡留司真)

5.3 データセット

データセットとして、これまで発行された各社新聞や詰将棋雑誌「詰将棋パラダイス」に掲載された詰将棋約 14000 局から抽出した詰め手順を用いた。この詰将棋は全日本詰将棋連盟の荒川様からご提供いただいた。

5.4 評価について

私たちはこの評価プログラムを、詰将棋の良し悪しを定量的に評価するために作成した。まず私たちは、広瀬ら (1998) の論文を参考に、必要項目を洗い出すこととした。広瀬らの論文の中では、

- 探索局面数
- 駒取りの回数
- 駒の使用率
- 詰み局面における玉の開放度
- 玉の位置

といった要素が 12 個挙げられていた。その中から、グループでの話し合いの結果、以下の 7 つの要素に絞り込み、評価プログラムの作成を行った。

- 枚数
- 重み
- 玉の開放度
- 配置の広さ
- 駒取りの回数
- 駒の使用率
- 玉の位置

しかし、私たちがランダム法で作った詰将棋において、初期の持ち駒のみで玉方を詰ませるという事例が存在したため、

- 手数に対する持ち駒の数

という要素を加え、評価に差が出やすくなるように調整した。したがって、最終的には、8 つの要素を用いた評価プログラムとなった。

まずは、各項目の有用性について、説明する。

(※文責: 菊地勇然)

5.4.1 枚数について

この項目は、攻め方・玉方問わず盤面の駒の数を数えた値が使用されている。この項目が有効であると考えた理由としては、駒の数が少ないほど、詰め上がりの際に綺麗な盤面になることや、作意が適切に表れやすいという面で期待ができると考えたからである。したがって、この値は小さいほど評価の値をよくしなければならぬため、負の値をかけている。

実際に、この粗点に関して、全日本詰将棋連盟様よりご提供いただいたものと私たちがランダム法で制作した詰将棋とのそれぞれ平均を比較した際には、約 2 ポイント、全日本詰将棋連盟様よりご提供いただいた詰将棋の方がいいということが分かった。

また、Word2vec による駒の置き換えを行ったものを行う前のものを粗点同士で比較した際に

は、ほとんど差がなかった (1 ポイント以内の差しかなかった)。今回の置き換えでは、一手目の動きに対応できるように駒の位置を変えるため、盤面や持ち駒としてその駒が存在している場合には、元々の位置からずらすのみとなり、盤面の駒数に差が生まれないということが、時々起こるためだと考えられる。

(※文責: 菊地勇然)

5.4.2 重みについて

この項目は、それぞれの駒に評価値を持たせ、その中で、攻め方の評価値の和を負にし、玉方の評価値の和を正にした値を使用した。この項目が有効であると考えた理由は、単純に駒数だけでは判別できない盤面の有利・不利を可視化することができると考えたからである。また、この値は攻め方のポイントが高いほど値が小さくなるようにしており、この値には正の値をかけている。駒ごとの評価値は以下のとおりである。

- 歩兵 1 点
- 香車 3 点
- 桂馬 4 点
- 銀将 5 点
- 金将 6 点
- 角行 8 点
- 飛車 12 点

実際に、この粗点に関して、全日本詰将棋連盟様よりご提供いただいたものと私たちがランダム法で制作した詰将棋とのそれぞれ平均を比較した際には、約 6 ポイント、全日本詰将棋連盟様よりご提供いただいた詰将棋の方がいいということが分かった。

また、Word2vec による駒の置き換えを行ったものを行う前のものを粗点同士で比較した際には、約 1 ポイント程度の向上が見られた。置き換えを行うプログラムでは、新しく挙げられた手順候補を達成する際に、盤面や持ち駒から必要な駒を持ってくるが、その必要な駒を置く場所に既に駒があった場合には、その駒を上書きして置くことになっていたため、上書きされた分の重みが減ったものと考えられる。

(※文責: 菊地勇然)

5.4.3 玉の開放度

この項目に関しては、詰め上がりの局面における玉将の回り X マス (最大は 8 である) のうち、駒で埋まっているのが P マスのとき、開放度は X-P と定義する。また、玉将が盤面の端にあり、駒を置くことができないマスが存在するときには、そのマスの分だけ、X の値を引く。この項目が有効であると考えた理由は、枚数のところでも述べたように、詰め上がりの際に綺麗な盤面である可能性が高いことや、玉将の開放度が高いほうが詰ませるための難易度が高く、いい詰将棋になりやすいと考えたからである。したがって、この値には正の値をかけている。

実際に、この粗点に関して、全日本詰将棋連盟様よりご提供いただいたものと私たちがランダム法で制作した詰将棋とのそれぞれ平均を比較した際には、0.2 ポイント、私たちがランダム法で制

作した詰将棋の方がいいということが分かった。こうなった原因として、私たちの詰将棋では、作意に反した駒が生成の過程で除去しきれていないために、詰む局面になりやすく、玉将の周りに駒が多くない状況でも、詰将棋として成り立ってしまうということが考えられる。

また、Word2vecによる駒の置き換えを行ったものを行う前のものを粗点同士で比較した際には、2ポイントの向上が見られた。これは、置き換えを行うプログラムで、大駒を優先的に使うような仕様としていたため、玉将のごく近くに駒がなくても、飛車や角行などの大駒で、詰む局面を作り上げることができるようになってきているということが考えられる。

(※文責: 菊地勇然)

5.4.4 配置の広さ

この項目に関しては、盤面における駒の配置の広さについて調べたものである。広さの調べ方として、一番左上にある駒の場所と一番右下にある駒の場所を結んだ対角線と、一番右上にある駒の場所と一番左下にある駒の場所を結んだ対角線とを比較し、大きい方の値をこの項目の値とする。この項目が有効であると考えた理由は、駒が色々なところに配置されている場合、雑然としたまとまりのない詰将棋になりやすいと考えたからである。したがって、この値には負の値をかけている。

実際に、この粗点に関して、全日本詰将棋連盟様よりご提供いただいたものと私たちがランダム法で制作した詰将棋とのそれぞれ平均を比較した際には、1.2ポイント、実際に、この粗点に関して、全日本詰将棋連盟様よりご提供いただいたものと私たちがランダム法で制作した詰将棋とのそれぞれ平均を比較した際には、約2ポイント、全日本詰将棋連盟様よりご提供いただいた詰将棋の方がいいということが分かった。

また、Word2vecによる駒の置き換えを行ったものを行う前のものを粗点同士で比較した際には、ほとんど差がなかった(1ポイント以内の差しかなかった)。今回の置き換えでは、大駒を優先的に使う仕様ではあったが、大駒以外の駒が置かれていた場所に大駒を置いて玉への距離には直接影響をもたらさないという事例があったと考えられる。

(※文責: 菊地勇然)

5.4.5 駒取りの回数について

この項目は、詰め手順の中で駒を取る回数を数えた値が使用されている。この項目が有効であると考えた理由としては、以下の二つがあげられる。第一に、駒取りが手順に必要となれば、必要であることを知らない挑戦者は駒を取るべきかという問に対する試行、どの駒をどの手番で取るべきかという試行を行うことになるからである。第二に、駒取りが詰将棋において独自の立ち位置を有しているからである。前述の通り、本将棋とは異なる詰将棋のルールとして詰む際には持ち駒はないというものがある。ここから、もし本将棋なら駒取りの次の手で王の前に金を打てば詰む、といった状態でもその取らざるを得なかった駒が桂馬であったりすれば、詰将棋では桂馬を打つ間に王が逃げることにより手順も大きく変わっていくことがあり得る。この二つの理由のどちらも、駒取りの回数が多い方がより複雑で頭を使うことにつながるため、正の数のままとした。

実際に、この粗点に関して、全日本詰将棋連盟様よりご提供いただいたものと私たちがランダム法で制作した詰将棋とで粗点の平均を比較した際には、約1ポイント全日本詰将棋連盟様よりご提

供いただいた詰将棋のほうが良いということが分かった。また、Word2vecによる駒の置き換えを行ったものを行う前のものを粗点同士で比較した際には、2ポイントの向上が見られた。これは、自動生成で作られた不自然な手のつながりの部分をデータベースにある自然なもの置き換えた際に、置き換えたところから旧来の次の手につなぐための手が増えたためだと考えられる。また、学習対象のデータベースに手数が多いものが多く、それを学習できたため、長くなる手に置き換えることに成功したということも考えられる。

(※文責: 中野豪大)

5.4.6 使用率について

最初から配置されている駒が詰め手順で動かされたかを使用率として評価に加えた。これは、置かれている駒の効きの位置を一回認識するだけでよい問題より、手が進むに応じて幾つかの駒の移動しうる場所、あるいは王に攻め込むことができる方向が変化していくほうが面白いと考えたからである。この項目は、より多くの駒が使われるほど変化に富む盤面であるため、正とした。

実際に、この粗点に関して、全日本詰将棋連盟様よりご提供いただいたものと私たちがランダム法で制作した詰将棋とで粗点の平均を比較した際には、約14ポイント全日本詰将棋連盟様よりご提供いただいた詰将棋のほうが高かった。また、Word2vecによる駒の置き換えを行ったものを行う前のものを粗点同士で比較した際には、11ポイントの低下が見られた。これは、置き換えの行われた対象が置き換え前の手順では動いていた駒だったこと、もしくは元から盤面に用意されているものではなく、途中で打ち込み、その後動かす駒が増えたことによるものだと考えられる。今回は置き換えた結果低下となったが、この項目が適切であったかどうかは今回の結果のみで決めるべきではなく、さらに実験を重ねることにより解明を目指して行く必要がある。

(※文責: 中野豪大)

5.4.7 玉の位置

この項目に関しては、玉の位置に関して調べたものである。玉の位置の定義としては、単純に玉将のある段及び筋を用いた。この項目が有効であると考えた理由は、二つある。一つ目の理由は玉の初期配置が逃げやすさに影響するからである。玉は一手につき1マスしか動かすことができないため、最初から端に追い詰められているか、中段に居て多くの逃げ場が残っているかで、詰め手順の手数も変わってくる場合がほとんどである。もう一つの理由としては、前述のように玉の位置が移動しても、相対的な詰め手順が変化しないものも存在する。そのような詰将棋ができた際に、評価の値に差を出すためこの項目を用意した。この項目の値は、中心に近づくほど大きくなる正の数とした。計算式は $9 - |(玉将のある段) - 5| + 9 - |(玉将のある筋) - 5|$ を用いた。

実際に、この項目に関して全日本詰将棋連盟様よりご提供いただいたものと私たちがランダム法で制作した詰将棋の平均の値を比較したところ、段の粗点は約1ポイント、筋は約2ポイント全日本詰将棋連盟様よりご提供いただいた詰将棋が下回った。この理由として、5.4.7配置の広さの項で述べた配置の広さが関係していると考えられる。配置が広いと、上下もしくは左右の離れたところに駒があるということになる。これが今回の結果につながったと考えられる。

また、Word2vecによる駒の置き換えを行ったものと、行う前のもので粗点を比較したところ、筋が2ポイント上回ったが段が2ポイント下回ったため、合計に変化はなかった。原因として、置

き換えの際にこの項目を重要視しない計算式にしたことが考えられる。

(※文責: 中野豪大)

5.4.8 手数に対する持ち駒の数について

この項目に関しては、詰め手順における攻め方の手番の回数と最初の持ち駒が何枚あるかを比較し、定めたルールに沿って一定の点数を評価に加えた。この項目が有効であると考えた理由を以下に述べる。1.1.1 詰将棋の概要の項で述べたように、持ち駒なく詰まなければならないルールがあることから、詰将棋では最初から持ち駒が与えられているとき、それを使い果たさなくてはならない。他にも、王手をかけ続けなければならないというルールもある。これも、長い手数の詰将棋を解くことを難しくさせる。また、ルールで規定されてはいないが、手数が増えるということは考えることができるので、挑戦者のミスが起きやすくなる。このようなルール下で挑戦者が問題を解く際には、盤上の自分の駒を指すことより、まず自分の持ち駒をできるだけ早く消費し、詰む方法を探すと考えられる。この際、攻め方の手番の回数と持ち駒が等しくなっているなら、その問題は打ち込みのみで解ける問題であるため、前述した思考方法のうちに解けてしまう。これに対し、攻め方の持ち駒より手番の回数のほうが多ければ、前述した思考の部分以上の思考が必要になる。すなわち、攻め方が打ち込みだけではなく盤上の駒を動かすこと、もしくは攻め方が玉方の駒を取って持ち駒とし、打ち込むことが必要となる。ここから、より複雑で面白くなると考えたからである。評価の値は、前述のルールにより手番の回数より駒数のほうが多いことはあり得ないので、手番の回数から駒数を引き差があるとき、差に応じて変化する正の数とした。また、引いた時の差が一つであれば、取って打ち込むという手はルール上あり得ないので、より簡単に詰め手順に辿り着くことができる。そのため、差が二つ以上の時よりもポイントの増加は少なくした。

実際に、この項目に関して、全日本詰将棋連盟様よりご提供いただいたものと私たちがランダム法で制作した詰将棋とで粗点の平均を比較した際には、約 23 ポイント全日本詰め将棋連盟様よりご提供いただいた詰将棋のほうが良いということが分かった。この結果は他項目と比べても比較的差が大きく、反映するのに十分な項目であることが確認できた。また、Word2vec による駒の置き換えを行ったものを行う前のものを粗点同士で比較した際には、40 ポイントの向上が見られた。これは、学習対象であるデータベースから詰め手順を学習する中で、駒を打ち込むだけでなく、駒を指す、取るといった行為に自然につながる手順を学ぶことができたためだと考えられる。また、5.4.2 重みの項で述べたように、置き換えの際に上書きして置くようにしているプログラムも原因の一つであると考えられる。

(※文責: 中野豪大)

以上の要素を用い、評価プログラムの値を算出する。各要素を以下のように置いた。

- 枚数 X_1
- 重み X_2
- 玉の開放度 X_3
- 駒取りの回数 X_4
- 駒の使用率 X_5
- 玉の位置 X_6
- 配置の広さ X_7

AI Love Deep Learning

- 手数に対する持ち駒の数 X_8

また、それぞれに対する係数を C_i とし、 i で対応したものにかけ合わせる。

- 枚数 $C_1 = 5$
- 重み $C_2 = 12$
- 玉の開放度 $C_3 = 1$
- 駒取りの回数 $C_4 = 12$
- 駒の使用率 $C_5 = 50$
- 玉の位置 $C_6 = 1$
- 盤面の広さ $C_7 = 36$
- 手数に対する持ち駒の数 $C_8 = 1$

上の値を用い、式に表すと以下のようなになる。また、各項目で負の値をかけると宣言した値には、この段階で負の値をかけている。

$$eval = -C_1X_1 + C_2X_2 + C_3X_3 + C_4X_4 + C_5X_5 + C_6X_6 - C_7X_7 + C_8X_8$$

(※文責: 菊地勇然)

類似手	Cos 類似度
3 八銀	0.602
2 八銀打	0.576
1 七金	0.572

表 6.1 2 七銀の類似手

類似手	Cos 類似度
2 八金	0.639
1 七玉	0.635
2 八飛	0.593

表 6.2 2 七金の類似手

類似手	Cos 類似度
8 四銀	0.525
9 二飛成	0.478
9 四香	0.476

表 6.3 9 三銀の類似手

類似手	Cos 類似度
3 八銀	0.598
2 八銀打	0.573
1 七金	0.560

表 6.4 9 三金の類似手

(※文責: 岡留司真)

6.2 詰将棋生成の結果

ランダム法の詰将棋生成プログラムによって詰将棋を作成する。その詰将棋の詰め手順を Word2vec 学習済みモデルに入力することで、詰将棋における類義語が出力される。ランダム法で作成された詰め手順を出力された詰め手順に入れ替えることで詰将棋を生成する。生成された詰将棋を評価関数に入力し評価値を出力する。

実際にランダム法で生成された詰将棋の例を図 6.2 に示す。この例での詰将棋は、7 手詰めの詰将棋で、持ち駒を打ち込んでいく単調な詰め手順となっている。その詰め手順は図 6.3 のようになっている。この詰将棋の詰め手順を Word2vec 学習済みモデルに入力させ、類似する詰め手順を出力する。出力された詰め手を入れ替えた結果、図 6.4 のような詰将棋が生成された。この詰将棋は 13 手詰めの詰将棋である。詰め手順は図 6.5 に示す。この詰将棋の評価値は-232 である。

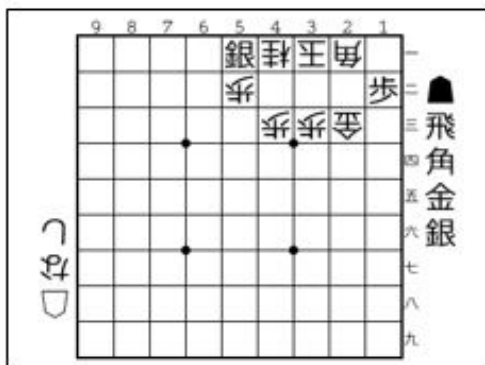


図 6.2 ランダム法で生成された詰将棋



図 6.3 ランダム法で生成された詰将棋の詰め手順

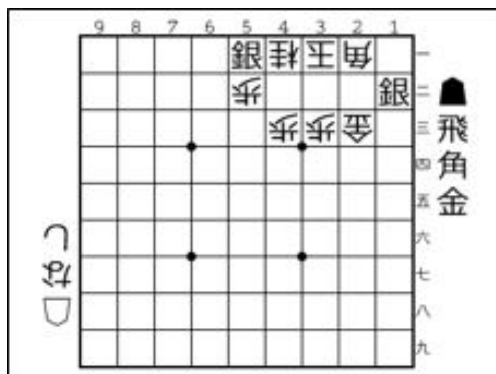


図 6.4 詰め手が入れ替えられた詰将棋の盤面

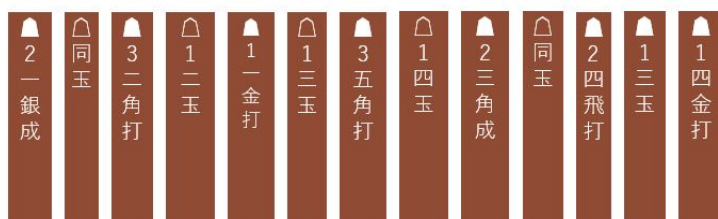


図 6.5 詰め手が入れ替えられた詰将棋の詰め手順

(※文責: 稲田将文)

第 7 章 考察

今回ディープラーニングを用いて手の入れ替えを行ったところ、手数が増加と、評価値の上昇が見られた。手数の増加については、ディープラーニングによって、大駒を使う手や、打ち込み以外の手に入れ替える確率が高くなったことが原因だと考えられる。学習に用いた詰将棋群に、大駒を使った手や、打ち込み以外の手が多く含まれていたため、これを学習したものと考えられる。評価値の上昇については、駒を入れ替えたことによる持ち駒の減少と、前述した通り大駒を使う手が増えたことが原因だと考えられる。評価値の高さが、面白さと考えると、今回のディープラーニングを使った詰将棋生成は、面白い詰将棋を生成する効果的な手法であるといえる。

(※文責: 渡邊凛音)

第 8 章 今後の展望

現状の詰将棋生成プログラムには、解決すべき課題が3つある。1つめは、盤面の生成である。今回、我々はランダム法を再現しようとしたが、完全にランダムで配置すると、ひとつの詰将棋を作るのに、莫大な時間がかかってしまうという問題がでてきた。そこで、攻め方の持ち駒を固定するという、苦肉の策をとったが、これでは、配置が限られてしまう。また、持ち駒は人の手により決められているため、完全な自動生成とは言えないものとなっている。今後の展望としては、盤面の生成にも、ディープラーニングを使うということがあげられる。ディープラーニングを用いて、コマの配置を確率的に考えることができれば、試行すべき配置を減らすことができ、生成にかかる時間を減らすことができると考える。さらに、駒の動きを時系列と捉え、CNNなどの手法を用いて学習させることで、面白い詰将棋の動きを学ぶことができると考える。また、現状のプログラムでは、盤面を 9×9 の配列で表し、値を駒の種類を表す0~31の数字として考えている。このままでは、Chainerで使いづらいため、配列を $9 \times 9 \times 32$ の3次元配列として、駒の種類を表す次元を増やし、駒が存在するかどうかを、0.1で表すようにすることで、Chainerでの学習をやりやすくすることができると考える。2つめは、駒の入れ替えである。現状のプログラムでは、駒の入れ替え候補の出力しか出来ないため、実際に入れ替えるのは人の手で行った。そのため、入れ替え後の棋譜の汎用性が失われてしまっている。今後の展望としては、入れ替え前と、入れ替え後の棋譜を学習させ、入れ替えの傾向を学習させることで、改善をすることがあげられる。

(※文責: 渡邊凜音)

第9章 まとめ

私達のグループはディープラーニングを用いて詰将棋の自動生成することを目的としている。既存の方法では単調でつまらない詰将棋を生成してしまう問題点がある。そこで、ディープラーニングを用いて、人間が作った詰将棋の特徴をうまく抽出し従来の自動生成よりも面白い詰将棋ができないかと考えた。

前期の活動期間中は、ディープラーニングの画像認識と自然言語処理の手法を応用し実装することを検討していた。しかし、多くの試行の結果、自然言語処理による手法を応用する方が有効であると考えた。詰将棋を自然言語処理で学習するために、私達は詰将棋を解く手順である詰め手順に着目した。自然言語処理では文章と単語の集合体であるコーパスを用いて学習させる。詰将棋における文章は詰め手順に相当し、単語は詰め手に当たる。詰将棋を学習させるに当たって採用した自然言語処理の手法は Word2vec である。Word2vec は単語の意味推測に用いられ、詰将棋の学習に親和性があると思われた。Word2vec の手法を応用すれば、詰将棋の学習が出来るのではないかと考えた。詰将棋を学習させるためのデータセットとして全日本詰将棋連盟に提供して頂いた 14000 局の詰め手順を使用した。

しかし、詰将棋を生成することはディープラーニングで詰将棋を学習した Word2vec 学習済みモデルではできない。そのため、私達はランダム法による詰将棋生成プログラムの開発も行った。従来のランダム法を参考にしプログラムの作成した。

また、私達は詰将棋を評価するために評価プログラムを作成した。評価する要素は枚数、駒の価値（重み）、玉の開放度、駒取りの回数、駒の使用率、手数に対する持ち駒の数となっている。これらの要素に適切な係数を掛け合わせ、評価の関数の値とした。

学習の結果は詰め手の分散表現をグラフとして可視化し、類似する詰め手を出力することに成功した。

ランダム法で詰将棋を生成し、ディープラーニングを用いて、手の入れ替えを行ったところ、手数の増加と、評価値の上昇が見られた。

手数の増加はディープラーニングによって大駒を使う手や打ち込み以外の手に入れ替える確率が高くなったことが原因だと考えられる。評価値の上昇については、駒を入れ替えたことによる持ち駒の減少と大駒を使う手が増えたことが原因だと考えられる。評価値が上昇したことを考えると、今回のディープラーニングを使った詰将棋生成は面白い詰将棋を生成する効果的な手法であるといえる。

この詰将棋生成の課題は、盤面の生成と駒の入れ替えである。盤面の生成は完全な自動生成ではなく、制限を設けるような形での生成となっている。盤面の生成にもディープラーニングを使うことで、完全な自動生成ができるのではないかと考える。また、詰め手の入れ替えはプログラムではなく手動で行った。入れ替え前と、入れ替え後の棋譜を学習させ、入れ替えの傾向を学習させることで、改善できるのではないかと考える。

(※文責: 稲田将文)

参考文献

- [1] K.Noshita, A Note on Algorithmic Generation of Tsume-Shogi Problems. Game Programming Workshop in Japan '96, 27-33, 1996.
- [2] 広瀬正幸, 伊藤琢巳, 松原仁. 逆算法による詰将棋の自動創作. 人工知能学会, 1998.
- [3] 新納浩幸, Chainer による実践深層学習, オーム社, 2016.
- [4] 斎藤康毅. ゼロから作るディープラーニング. O'Reilly Japan. 2016.
- [5] 清水亮. はじめての深層学習プログラミング. 技術評論社. 2017.