

# 命題論理の深層学習による自動証明

金原 雅典<sup>†</sup> 佐藤 亮介<sup>†</sup> 鷗林 尚靖<sup>†</sup> 亀井 靖高<sup>†</sup>

<sup>†</sup>九州大学

## 1 はじめに

プログラムの正しさを証明することは、プログラムに誤りがないということを保証し、膨大なテストケースを検査したり余計なデバッグ作業を無くす、あるいは減らすことに貢献する。形式的に保証することは大変であり、その証明作業を自動化することができれば、多くのプログラムの正しさを証明する手間を大幅に減らすことができる。プログラムの正しさを証明することで、開発の効率化及びプログラムの信頼性の向上が見込める。

人間の知能や作業を代替する手法として、近年特に注目を集めているものの一つに機械学習手法がある。機械学習は、機械翻訳を用いた自然言語間の翻訳や、強化学習による囲碁プログラムなど、様々な分野で研究・応用され、多様な手法が考案されている。証明の分野においても、命題とその証明の関係も命題から証明言語への変換であると見ることができ、命題とその証明をトークン列で表現すれば機械学習を適用することで自動証明が可能になると考えられる。

本研究では、学習に使用するデータセットが十分に存在すればシーケンス変換モデル [3] で命題から証明言語への変換が学習可能かどうかということを検証するために、命題論理に限定した論理の機械学習及び自動証明を行う。命題論理の証明のデータセットをプログラムで作成し、シーケンス変換モデルに学習させた。

本研究での“証明”は自然言語により書かれたものではなく、定理証明支援システム Coq [1] 上で書かれたものである。ここでは、Coq の証明構築を機械学習モデルの一つであるシーケンス変換モデルを用いて学習させることを試み、その精度を確認する。

## 2 学習対象とする証明

本章では、学習対象とする証明について述べる。まず、本研究で用いた命題論理について定義する。次に、

### Automated theorem proving of propositional logic by deep learning

Kanahara Masanori<sup>†</sup>, Sato Ryosuke<sup>†</sup>, Ubayashi Naoyasu<sup>†</sup>, and Kamei Yasutaka<sup>†</sup>

<sup>†</sup>Kyushu University

kanahara@posl.ait.kyushu-u.ac.jp

{sato, ubayashi, kamei}@ait.kyushu-u.ac.jp

その“証明”を構築する Coq について述べる。

### 2.1 命題論理式の定義

本研究では学習のデータセットとして、直観主義論理において恒真である命題論理の証明を用いる。命題論理を用いた理由は、その論理の構造が単純であり、機械的に多くのデータセットが作成可能であるためである。対象の命題論理式は以下の BNF 記法で表せる。

$$f ::= x \mid \text{True} \mid \text{False} \mid f \wedge f \mid f \vee f \mid f \rightarrow f$$

$x$  は変数、 $f$  は命題論理式である。結合子は論理積  $\wedge$ 、論理和  $\vee$ 、含意  $\rightarrow$  の 3 つを用いる。ここでは否定は対象の論理を単純化するため対象としないこととする<sup>1</sup>。また、命題論理の複雑さを表すものとして構文木の深さを用いる。

### 2.2 Coq

本研究では定理証明支援システム Coq 上での証明を対象として学習を行う。Coq とは定理証明支援システムであり、証明作業を支援し正しさを保証するツールである。Coq は自動で証明を行ってくれるといったものではなく、あくまで人の証明作業を手助けするツールである。Coq は高階述語論理を扱うことができるため、多くの論理を表現できる。証明構築は Tactic と呼ばれる Coq に対する計算命令を対話的に適用していくことで、Coq 上の記法で記述された証明の副目標である「ゴール」や「仮定」を変化させていき、命題の正しさを保証することができる。証明を開始した時点でのゴールは証明すべき命題そのものになる。Tactic を適用させていった結果、「証明すべきゴールがない」という状態になれば証明終了である。Tactic が適用できない場合、Coq はエラーを返す。

証明を構築する上ではどのように証明するか、つまりどの Tactic を適用するかは証明者が選択する必要がある。適用する Tactic の選択は命題が複雑なものになるに連れて難しく手間のかかるものになる。適用すべき適切な Tactic を自動で推論することができれば、ゴールや仮定を変化させるなどの作業は Coq 自体が行なっ

<sup>1</sup>否定  $\neg P$  は  $P \rightarrow \text{False}$  と表すことができる。

表 1: 結果一覧

記法	前置	中置	後置
正答率	96.3 %	95.6 %	95.9 %

てくれるため、証明構築全体を自動化することができるようになる。

### 3 実験手法

学習にはシーケンス変換モデルを用いる。シーケンス変換モデルとは、入力のシーケンスと出力のシーケンスの組を学習させ、入力を受け取ったとき、その入力を元に推論した出力を返す学習モデルである。ライブラリは TensorFlow<sup>2</sup> の seq2seq を使用する。

本研究では仮定とゴールを順に並べたものを入力として、その仮定とゴールに適用できる Tactic を出力としてデータセットを用意し、それを用いて学習させる。学習データはプログラム上で変数の個数と命題論理の深さを限定し、恒真式の命題論理を生成した。また、そのゴールと仮定が証明完了に向かうために適用すべき Tactic を 1 つ作成した。プログラムで生成した「仮定とゴール」と「適用すべき Tactic 1 つ」をデータセットの入力・出力の組とする。今回の検証では変数の個数は 3、深さを 5 としてプログラム上で生成を行い、学習用データを 100,000 件生成し、それらを train 用の 80,000 件と validation 用の 20,000 件に分割した。また、同様に結果確認用のデータを 20,000 件の生成を行った。

また、命題論理の演算子の位置によって差異があるかどうかの確認を行うために、前置・中置・後置それぞれのデータセットを作成し、それぞれ学習を行った。

### 4 結果

結果は図 1 のようになった。おおよそ 95 % の精度で正しく推論できることが確認できた。また、前置・中置・後置等の演算子の位置による違いはほとんどないことが確認できた。

関連研究の 1 つである Sekiyama らの研究 [2] では、証明の各ステップを独自の機械学習モデルを用いて学習させている。学習結果としては、深さが 5 での全ての推論規則では 96.92 %、深さ 1 から 26 までの全体を通してでは 96.79 % の精度を示したと述べている。

Sekiyama らの研究と本論文では深さの定義が異なる

ため単純な比較はできないが、証明の 1 ステップごとに分割した精度ではほぼ同等の精度を出している。証明開始から証明終了まで全体を通しての精度を検証することは今後の課題である。

### 5 まとめと今後の予定

本研究では、命題論理に対象を限定した命題の自動証明が可能かの検証を行うために、Coq 上の命題論理の恒真式の証明に対して機械学習モデルのシーケンス変換を用いて学習を行い、その評価を行った。結果としては、演算子の位置に寄らず約 95 % の精度で正しく推論できることを確認した。

今後の発展として、証明開始から証明終了まで通して推論させた時にどの程度推論できるかという検証や、データセットの量によって精度がどのように変化するかを検証を行っていく予定である。また、より精度を向上させるためにはどのようにしたらいいかということや、命題論理だけではなくそれ以外の論理にも適用する上での問題を解決するためにはどうすればいいかを議論していく必要がある。

命題論理の自動証明ができると確認できた場合、命題論理だけではなく一階述語論理へと学習範囲を拡張する。学習に用いるデータセットは Coq の標準ライブラリや、Coq 開発陣以外が証明し一般に公開されている Coq のライブラリ<sup>3</sup>を用いて作成する予定である。

#### 謝辞

本研究は、JP26240007, 18H04097 による助成を受けた。

#### 参考文献

- [1] development team, T. C.: *The Coq proof assistant reference manual*, 8.7.1 edition (2017).
- [2] Sekiyama, T. and Suenaga, K.: Automated Proof Synthesis for the Minimal Propositional Logic with Deep Neural Networks, *Programming Languages and Systems - 16th Asian Symposium, APLAS 2018, Wellington, New Zealand, December 2-6, 2018, Proceedings*, pp. 309–328 (online).
- [3] TensorFlow: Sequence-to-Sequence Models, <https://www.tensorflow.org/versions/master/tutorials/seq2seq#sequence-to-sequence-models>.

<sup>2</sup><https://www.tensorflow.org>

<sup>3</sup><https://coq.inria.fr/opam/www/>