Summer Research Program in Industrial and Applied Mathematics 2017

Towards Automated Theorem Proving: A Deep Learning Approach

Statement of Work

Sponsored by
Real AI, SNU, HKUST

Student Members
Gwang Hyeon CHOI
Seung Yong MOON
Zheng PAN
Jiaze SUN

Industrial Mentor
Jonathan YAN

Academic Mentor
Queenie LEE

June 30, 2017

1 Introduction

1.1 Company Background

Real AI is a company based in Hong Kong, and as its name suggests, focuses on the research and development of safe and beneficial artificial general intelligence. Adopting the principle of effective altruism, Real AI aims to develop altruistic intelligence, which essentially is an AI that can contribute as much as it can in solving many of today's problems, such as disease, climate change, or even poverty.

1.2 Problem Background

Automated Theorem Proving (ATP) is about proving mathematical statements using computers, or more precisely, artificial intelligence. The task of constructing proofs is an arduous one, since it usually requires substantial experience and knowledge in the discipline. A human mind might need years and even decades of intensive training to be able to successfully produce proofs, but with the help of deep learning and modern computers, it is possible to let an AI "learn" mathematics and generate proofs much more efficiently.

In order to accomplish such a task, we must firstly formalise the mathematics that we humans understand into one that can be processed by the computers, and this is where formal maths systems come into play. Within such a system, every step of a proof follows directly from the previous one, and all theorems can be verified algorithmically. Various datasets of proofs using formal maths systems include Metamath, Mizar and HOL light.

2 Holophrasm: A Technical Overview

Holophrasm is a neural automated theorem prover for Metamath written in Python. It utilises a neural-network-augmented bandit algorithm as well as a sequence-to-sequence model for action enumeration to the partial proof trees in Metamath language. Currently, Holophrasm can provide correct proofs to 14% of the test theorems in Metamath's set.mm database.

2.1 Metamath

As previously mentioned, a formal maths system is the foundation of ATP, and Metamath, meaning "metavariable" math, is such a formal language. It includes over 19,000 proofs, all of which can be traced back step by step to a list of 22 axioms, which constitute the foundation of mathematics known as Zermelo-Fraenkel set theory.

The process of proving theorems in a formal system involves a very straightforward technique – substitution, which is essentially the application of a more general law to a more specific scenario. Let T be a theorem that has been proven, and denote by a_T, e_T and f_T respectively its assertion, its hypotheses, and its free variables that appear in the assertion and hypotheses. As an example, consider an assertion a that is to be proven. In order to apply T in proving a, a set of substitutions, ϕ, must be applied such that $\phi(a_T) = a$. In doing so, we have reduced the problem of proving a into a problem of proving $\phi(e_T)$.

Apparently, the substitution cannot be carried out casually. For instance, there are different types of variables within the Metamath system, and the variables which replace the free variables must be of the corresponding type, otherwise it is not a proper substitution, and will result in an incorrect proof.

2.2 Proof Tree

Based on the preceding discussion, we can see that the process of constructing proofs naturally gives rise to a tree structure, which consists of two different types of nodes, red and blue. Red nodes represent expressions, which can be assertions or hypotheses; and every blue node represents the application of a theorem, and thus labeled by the name of the theorem and a set of substitutions. Suppose we have a context proposition C with a set of hypotheses e_C and assertion a. First, the root is set as a red node and labeled by a. A red node has exactly one child, a blue node, which represents the application of a theorem T. Intuitively, only one theorem is required to "break down" an assertion into a number of hypotheses. Unlike a red node, a blue node has the same number of children, which are red nodes, as the number of hypotheses in T, since a theorem can only be applied when all of its hypotheses are satisfied. The process goes on until all the leaves of the tree are elements in e_C. If such a tree exists, it is called a proof tree.

Upon given an proposition, Holophrasm will attempt to search for a proof tree within the search space called a partial proof tree, which is an extension of the proof tree. In a partial proof tree, a red node is permitted to have numerous blue node children, and it is also allowed to have no children even if it is not a hypothesis. A red node is called proven if at least one of its child blue nodes is proven, and a blue node is proven if all of its child red nodes are either proven or in e_C.

2.3 Deep Learning

Deep learning is a part of Machine learning, which uses Artificial Neural Network (ANN) for implementing machine intelligence. The most important feature of deep learning is multiple hierarchical levels of data features, where the features of higher levels are defined in terms of lower ones. There are many applications of deep learning with ANN, such as Recurrent Neural Network(RNN), Convolutional Neural Network(CNN). Deep learning is showing great efficiency in various fields, especially in computer vision and natural language processing.

The problem of proving a proposition amounts to finding a proof tree within an essentially infinite search space. Hence, deep learning is implemented to tackle a task. Holophrasm has 3 neural networks, which are the payoff network, relevance network, and generative network. At a red node, the relevance network guesses which theorem can be used to prove the given expression, and the most promising one will be selected.

The generative network generates unconstrained substitutions until they cover all free variables necessary. At a blue node, the payoff network evaluates the payoff of all child red nodes. The least promising child red node is selected, since if it turns out that this red node cannot lead to a proof of the theorem, then the parent blue node can be abandoned.

3 Project Aim

The aim of the project is to gain a solid understanding of and implement the Holophrasm algorithm, and if possible, make improvements upon it.

4 Tentative Schedule

Week 2 – 3 :

- Study various academic papers and gain an understanding of the algorithm
- Go through the python codes of Holophrasm and recognize the functions in each steps

Week 4 – 5:

- Implement the Holophrasm algorithm
- Improve Holophrasm by debugging existing code and if possible, incorporating TensorFlow, which allows faster training and easier implementation of techniques such as convolutional neural network (CNN)

Week 6:

- Keep working on the algorithm
- Prepare for the final presentation

Week 7:

- Finalise the product
- Final presentation and report

5 Deliverables

- Final presentation
- Final report
- An improved version of Holophrasm

References

[1] Whalen, D.P.Z.*Holophrasm: a neural Automated Theorem Prover for higher-order logic*, `https://arxiv.org/pdf/1608.02644.pdf`, 2016

[2] Silver, D., Huang, A., ..., Hassabis, D.*Mastering the game of Go with deep neural networks and tree search*, 2016