Proof Prediction Deep Learning Project Executive Summary

Team: Jared Able, Evgeniya Lagoda, Hongyi Shen, Dennis Nguyen, Zhihan Li

Github: jableable/proof-truth: Erdős Institute - Summer 2024 - Deep Learning Bootcamp Project (github.com)

Overview:

In recent years, generative AI based on deep neural networks has become a multi-billion (soon perhaps trillion) dollar industry. A major application of this technology is chatbots which have been developed to provide information to consumers. Notably, chatbots can mimic human conversation, but they struggle with logical reasoning and hallucinations. Our project applies deep learning techniques to mathematical logic to discover the limits and strengths of teaching a model to perform logic.

A major goal of AI developers is to develop technologies that produce accurate and logical information, and thus applications of machine learning techniques to formal logic are an important test case. Regulators, policymakers, and consumers of technology also benefit by knowing the potential and limits of neural network based technology. Several primary uses of AI, for example problem-solving and generating computer code, heavily rely on rigorous and impeccable logic.

Our overall goal was to develop a model that can predict a single proof step at a time by using deep learning methods. We developed models to predict both the statement of a given proof step as well as its logical justification (aka its label). We also made a model to predict the distance between two statements within a proof tree.

Our Metamath Dataset:

Our dataset consists of 42,494 theorem proofs written in the Metamath computer language. Metamath is a simple logic system accompanied by a command-line tool for proving and verifying theorems. We wrote an API to extract the steps of the proofs and convert them into a graph format. Each theorem corresponds to one (directed) graph, where nodes are proof steps and edges connect the nodes when there is a dependency in the theorem steps.

Label Prediction (GIN model):

We exploited the graph structure of our proofs by applying a Graph Isomorphism Network, also known as a GIN, to predict the logical step needed to reach a given node. In mathematics, this is a challenging problem because there are often many valid ways to combine assumptions into a conclusion. In our setup, there are over 2,000 logical steps to choose from.

To train our model to pick the correct logical step, we began by transforming each node's statement into a vector embedding by using Google's Universal Sentence Encoder. Our GIN model then takes in these statement features as inputs, and as its output, it predicts the logical step required to reach a given statement. While we train our model on the entire graph, we specialize our application to predicting the final conclusion node's label without knowing

the conclusion node's statement. In other words, the model must rely entirely on nearby nodes to predict the conclusion node's label.

We capture a successful prediction with not just the top choice produced by the model, but with the top five choices. This Top 5 accuracy, when restricting to the conclusion node, is about 70%. Considering that there are over 2,000 options to pick from when making a prediction, this model performs very well.

Statement Prediction (LSTM RNN model):

The model for generating statements uses a combination of graph random walks and language processing models. The overall goal of this model is to predict the statement of the penultimate step based on the previous steps. The architecture is a long short-term memory (LSTM) recurrent neural network with a single hidden layer. We used random walks up the proof tree to generate blocks of text. These blocks of text were converted to skip-grams on which the model was trained. Statements are predicted character by character.

The results were evaluated on a subset of the proofs: propositional logic without quantifiers. The performance of this model is mediocre. However, the two largest challenges were inherent to the question, not the model itself. In some cases, the penultimate statement is an assumption and thus difficult to predict. The second problem is that, while the model predicts logically valid statements, these statements are, however, not the desired conclusion. In most cases, when the statement did not approach the length of the skip-grams, the structure of the statement was recognizably similar to the correct statement. In almost all cases, the model produces a valid statement.

Distance Prediction (Attention-Based Model):

We developed an attention-based model to assess how closely a given statement relates to the desired conclusion. This approach simulates the intuition mathematicians use when determining whether a particular deduction is valuable for proving the conclusion. In this model, we first define a distance function, which measures the number of statements separating a given statement from the conclusion within the proof tree, after decomposing the tree into simpler subtrees.

The model is constructed by treating all leaf nodes—embedded and vectorized—as the keys in the attention mechanism, with the conclusion serving as the query. The attention mechanism, combined with an intermediate statement, is then used to predict this distance. After 100 epochs, the MSE loss decreased from 7.70 to 2.76.