Symbolic Manipulation and Computation in the Same Graph

Darius Barbano

Contents

Abstract

General artificial intelligence refers to machine intelligence than performs a task as successfully as a human does. A fundamental difference between human neural network and current machine neural networks is that only human networks combine symbolic reasoning with computation.

1 Background

Neural networks have been used for decades to solve a wide variety of machine learning tasks without the need for explicit programming of the solutions to be done by humans. Typically the networks only make use of arithmetic computation, meaning that logical operations are not incorportated into the models. This hasn't been done because there is no clear method by which logical information (TRUE/FALSE) should be converted to numerical information, and vice versa. In Computer Science, True and False are generally represented as 1 and 0, respectively, whereas in Mathematics, they are often represented as -1 and 1 Which specific fields of Math and Compsci?. Our research aims to construct a neural network which effectively combines logical and arithmetic computation, and apply this network to a problem which demostrates it's ability to perform symbolic reasoning.

1. What is symbolic computation?

   A symbolic computation is a calculation performed with symbolic representations of values and operations. A simple example would be the expression $(x+1)(x-1)$ which would evaluate to x^2-1, rather than to some numerical result.

2. What is calculation?

   A calculation is a process by which one or more inputs is transformed into one or more results. One may calculate that the product of 5 and 4 is 20.

3. What is meant by a computational class? - do you mean complexity class?

Code for each subsection of this document can be found at NeuralNetworkResearch.

2 Methods

2.1 Network Construction

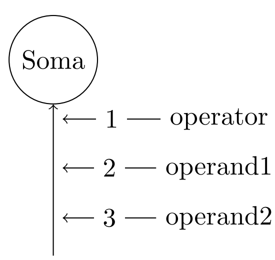

Figure 1: Single neuron receiving ordered intputs.

While this model is suitable for biological networks of neurons, a neural network in the computational sense would look more like an abstract syntax tree. In the following diagram, a network of operators and operands form a tree-like structure in which elements at lower levels represent values to manipulate which at higher levels interact with each other through arithmetic operations to produce a result. Any computation which involves a combination of operations on one or more values can be represented in this structure, such as logical or arithmetic operations.

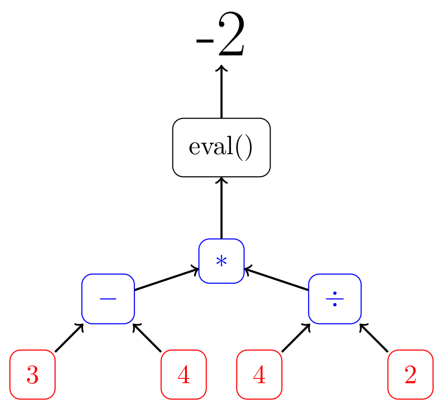

Figure 2: An Abstract Syntax Tree of arithmetic operations.

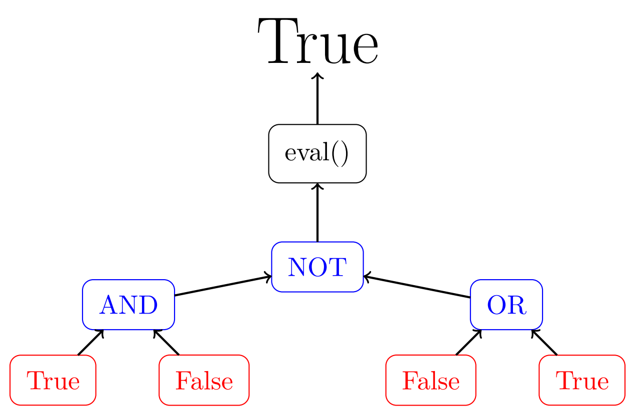

Figure 3: An Abstract Syntax Tree of logical operations.

Syntax trees such as the ones shown above work perfectly, since each contain exclusively arithmetic or logical computations. In the trees shown below, however, the two computation classes are combined through multiplication and addition. The tree's results depend entirely on how we choose to evaluate $2 * True$ and $2 + True$.

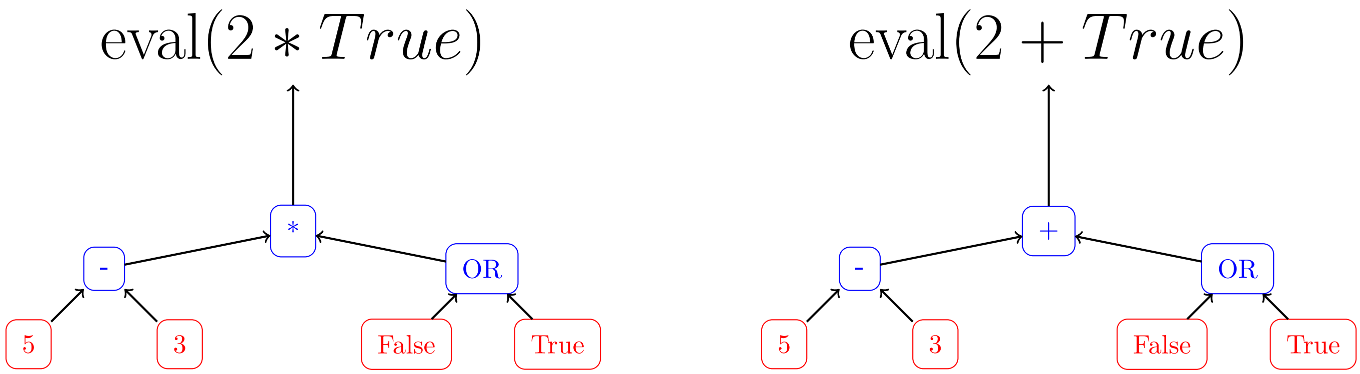

Figure 4: Abstract Syntax Trees of logical and arithmetic operations.

3 Results

4 Conclusions