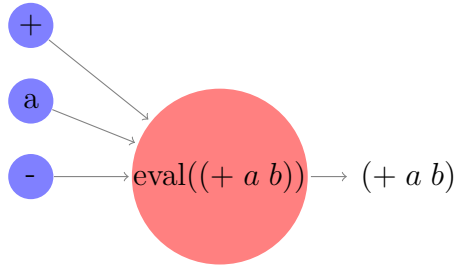


Consider the construction of a neural network to alternately add and subtract numbers. It is useful to represent all binary operations in list form,  $(+ a b)$ . The neural network, then, becomes a representation of a stack.

Input

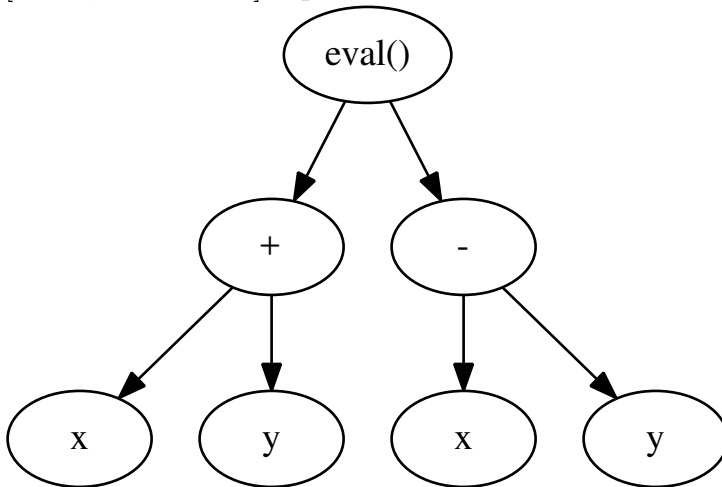


```

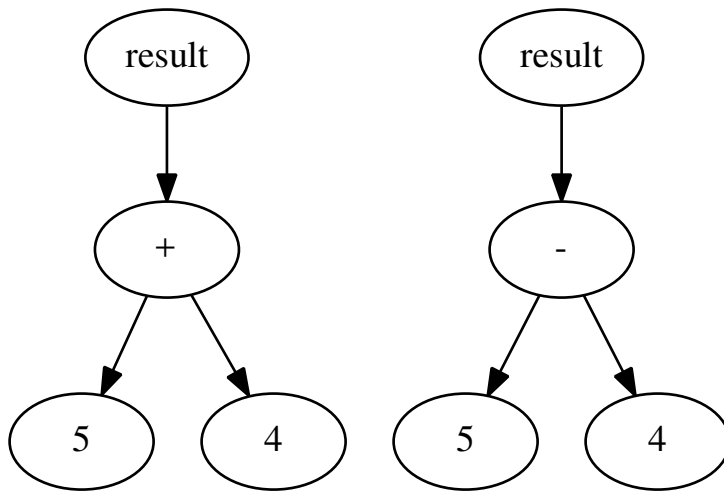
;;;
yshift=0.5cm]nodehidden neuron] (H-1) at (2.5cm,-1cm) +;yshift=0.5cm]nodehidden neuron] (H-2)
at (2.5cm,-2cm) a;yshift=0.5cm]nodehidden neuron] (H-3) at (2.5cm,-3cm) -;
[output neuron,pin=[pin edge=-,]right:(+ a b), right of=H-3] (O) eval((+ a b));
H-\source)edgeO);H-\source)edgeO);H-\source)edgeO);

```

[annot,left of=H-1] Input;



This tree-like structure closely mimics the abstract syntax tree which is generated by compilers when processing code in a program. For example, to evaluate the results of the expressions  $+ 5 4$  and  $- 5 4$ , a compiler may generate the following trees:



To improve this sentence don't use *weasel words*. I don't know what *very similar* means to you. Neither a compiler nor interpreter *view* code. Neither are (yet) sentient beings.

The best piece of writing advice I can give is:

Mean exactly what you write and write exactly what you mean.