Logical Neural Networks: Opening the black box

Boolean Like Neuron Paramaterisation

Daniel Braithwaite

April 27, 2017

1 Background

Part of the issue with the CL approach was that when using these non standard functions with differentials that dident make sense in the case with many inputs the gradients dident work correctly.

The idea behind this approach is similar to what has been done, we want to design functions which we can easily differentate and when we fix inputs to 0 or 1 then the functions act like boolean functions. Further dicussion of these Boolean Like functions can be found in R. J. Williams (1986) [**?**]

2 Development of Paramaterisation

We have our inputs to the neuron $x_1, ..., x_n$ and for each input we have a corosponding weight $\epsilon_1, ..., \epsilon_n$. We think of the corosponding weight for an input as our belef that **input** x_i **does not effect the output**. In the case of our NAND neuron we consider the activation function to be $NOT(AND(OR(x_1, \epsilon_1), ..., OR(x_n, \epsilon_n)))$. We can intepret these logical functions as the folowing boolean like functions

$$NOT(x) = 1 - x \tag{1}$$

$$AND(x_1, ..., x_n) = \prod_{i=1}^{n} x_i \tag{2}$$

$$OR(x_1, x_2) = x_1 + x_2 - x_1 x_2 \tag{3}$$

With equations (1), (2) and (3) in mind our paramaterisation becomes

$$y = 1 - \prod_{i=1}^{n} (x_i + \epsilon_i - x_i \epsilon_i) \tag{4}$$

3 Experemental Results

This paramaterisation is able to learn NOT, NAND, AND, OR (if the network has the right configuration) however trying to learn something like implys demonstraites a key issue with this approach. Implys can be given by $p \implies q \iff p \uparrow (q \uparrow q)$ but our networks are feedfoward with the layers being fully connected, meaning one of our NAND neurons would have to act as as an identity to one of the inputs, which it cant do

References

[1] Williams, R. J. The logic of activation functions. *Parallel distributed processing: Explorations in the microstructure of cognition 1* (1986), 423–443.