

Logical Neural Networks: Opening the black box

Logical Normal Form Network

Daniel Braithwaite

May 9, 2017

Given a simple feed forward network using neurons resembling NAND gates is unable to learn any given boolean formula we switch our interest to something else. We know that any boolean expression can be represented in Conjunctive Normal Form (CNF) or Disjunctive Normal Form (DNF). So it seems prudent to ask whether it's possible to construct a feedforward neural network which can learn these underlying representations. Findings [1] seem to suggest that it is possible however there is limited justification for some claims and because of this is difficult to reproduce. We will take this general concept and reproduce the research in an attempt to develop a better understanding.

1 Noisy Neurons

During a previous investigation of Logical Neural Networks the concept of Noisy-OR and Noisy-AND neurons were derived [2]. Here we will simply state them as.

Definition 1.1. A **Noisy-OR** neuron is a perceptron with activation $a = 1 - e^{-z}$ where W is our weights, X is our inputs, b is our bias term and $z = WX + b$. We constrain each $w_i \in W$ and b to be in the interval $[0, \text{inf}]$

Definition 1.2. A **Noisy-AND** neuron is a perceptron with activation $a = e^{-z}$ where W is our weights, X is our inputs, b is our bias term and $z = WX + b$. We constrain each $w_i \in W$ and b to be in the interval $[0, \text{inf}]$

2 Logical Normal Form Networks

A Logical Normal Form Network is a neural net which satisfies one of the following definitions

Definition 2.1. CNF-Network A **CNF-Network** is a three layer network where neurons in the hidden layer consist solely of Noisy-OR's and the output layer is a single Noisy-AND.

Definition 2.2. DNF-Network A **DNF-Network** is a three layer network where neurons in the hidden layer consist solely of Noisy-AND's and the output layer is a single Noisy-OR.

It is worth noting that CNF and DNF form can have the notes of atoms in there clauses, a simple way to account for this is to double the number of inputs where one represents the atom and the next represents the negation of that atom.

3 CNF Networks

We begin our investigation of LNF networks by experimenting with CNF Networks, both on simple 2 variable gates and on more complicated functions

3.1 Learnability Of Boolean Gates

Theoretically it makes sense for these networks to be able to learn the CNF and DNF representations of various boolean expressions, however before we attempt arbitrary boolean functions we would like to start with something simple, namely expressions such as NOT, AND, NOR, NAND, XOR and IMPLIES. Results of which were promising, we were able to achieve a low error and from inspecting the weights we could see that the networks were in fact learning the correct CNF and DNF representations.

3.2 Learnability Of Interesting Expressions

We now wish to see if we can use these networks to learn more interesting boolean formula, starting with expressions of 3 variables. While we are able to achieve a small error there is now some noise (i.e. small non zero weights for inputs that are irrelevant). While a small amount of noise is okay and can be pruned out after training we could run into issues if the amount of noise increases as the number of inputs does.

References

- [1] HERRMANN, C., AND THIER, A. Backpropagation for neural dnf-and cnf-networks. *Knowledge Representation in Neural Networks, S* (1996), 63–72.
- [2] LAURENSEN, J. Learning logical activations in neural networks, 2016.