Logical Neural Networks

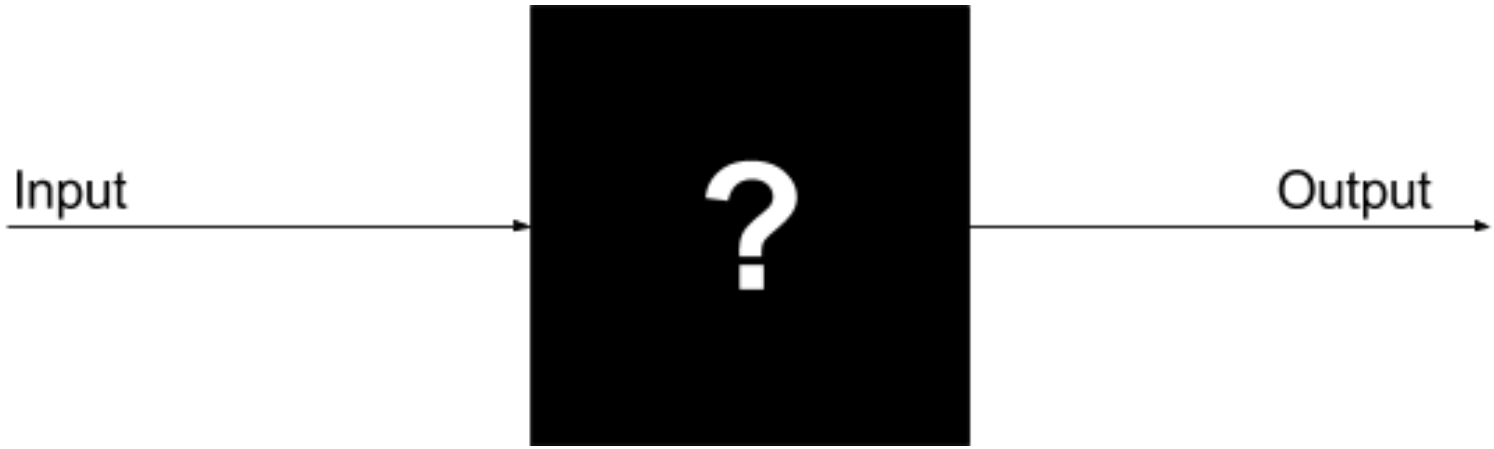
Opening The Black Box

COMP 489 Project

Daniel Braithwaite

Supervisor: Marcus Frean

Introduction + Motivation



Difficult to interpret Artificial Neural Networks using standard activations, e.g., Sigmoid, TanH.

Why Interpretable Systems?

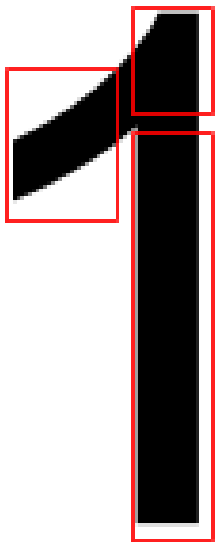
- Safety Critical Systems
- Ensuring systems make Ethical decisions
- European Union General Data Protection Regulation

Problem Statement

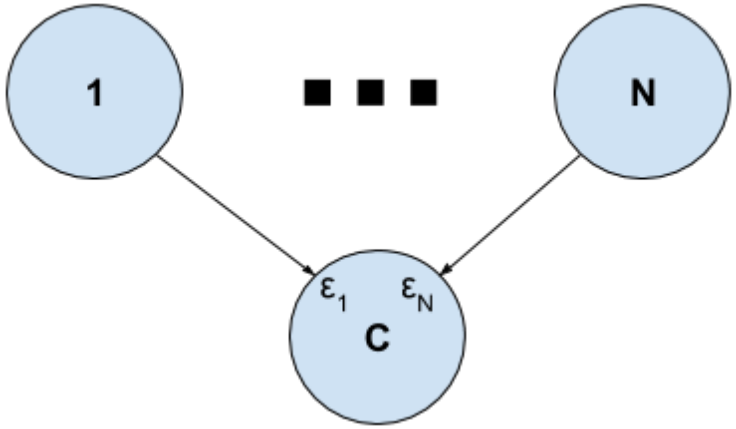
- Want Artificial Neural Networks which can achieve a high accuracy.
- Want Artificial Neural Networks which have an interpretable learned model so their predictions can be defended

Idea

- Some problems appear to have a logical decomposition
- Logical functions are a natural thing for humans to interpret
- **Goal:** Learn these logical decompositions using Backpropagation
- **Problem:** Standard Boolean Logic Gates are not continuous.

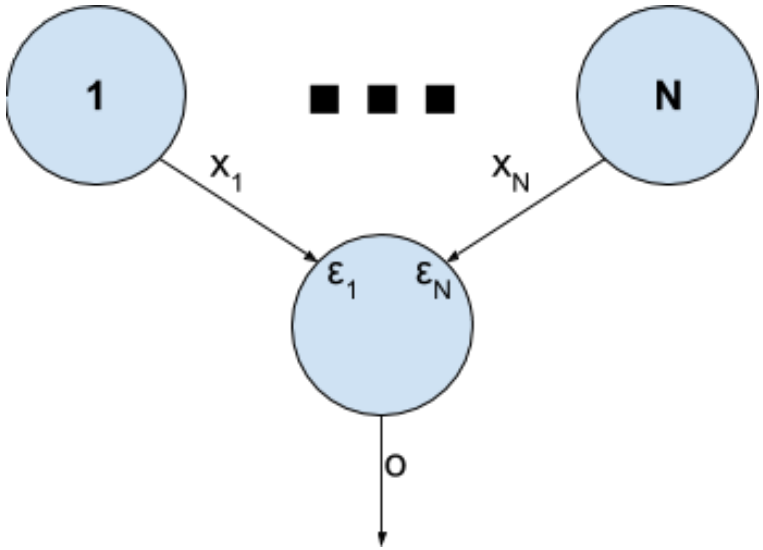


Noisy-OR Relation



- Every parent is on, but there exsists uncertanty as to node i influences the child.
- $\epsilon_i \in [0, 1]$ is the probability that input i is irrelevant to C.
- $C = OR(x_1, ..., x_n)$, so $P(C = 0|x_i = 1 \forall i) = 0$
- What if there is uncertainty that input i influences C. Then $P(C = 0|x_i = 1 \forall i) = \prod P(C = 0|x_i = 1)$
- Therefore $P(C = 1|x_i = 1 \forall i) = 1 - \prod \epsilon_i$

Noisy Neurons



- Noisy-OR relation almost gives the OR activation we are looking for.
- Insted each input node will be on with probability x_i.
- The total irrelevence of the node i is then defined as $\epsilon_i^{x_i}$
- The Noisy-OR activation is therefore $1 - \prod_{\forall i} \epsilon_i^{x_i}$
- In a similar fassion the Noisy-AND activation is given as $\prod_{\forall i} \epsilon_i^{1-x_i}$
- Both activations reduce to descrete gates when inputs are binary and $\epsilon_i = 0$.

Approach: Logical Neural Networks

Logical Neural Networks have layers consisting of Noisy Neurons. Can be trained with Backpropagation.

Problem: Weight Initialization

- Even small networks would not train.
- Derived a distribution from which to sample weights.
- Now large networks can be trained, including deep Logical Networks. Up to 10 layers deep were tested!

Experimental Approach

- Want to evaluate accuracy and interpretability of Logical Neural Networks
- Implement in Tensorflow.
- Logical Neural Networks are compared against Multi Layer Perceptron Networks (of equivelent size) using the MNIST problem.
- **Accuracy:** Networks trained from 30 different initial conditions, accuracy compared using confidence intervals obtained from evaluation of the network on a testing set.
- **Interpretability:** Results are obtained by visually comparing interpretations of the weights from different networks.
- It will be diffcult to give conclusive evidence given the experements are limited.

Experimental Results: Accuracy

- Logical Neural Networks have statistically equivalent accuracy to Multi-Layer Perceptron Networks.

Experimental Results: Interpretability

- Logical Neural Networks are potentially more interpretable that Multi-Layer Perceptron Networks.
- Interpretability of Logical Neural Networks depends on activations used.

Experimental Results: Interpretability - No Hidden

Pictures represent the weights in learned the models, specifically the output neuron representing a 0. Dark regions are most important, and white is irrelevant.

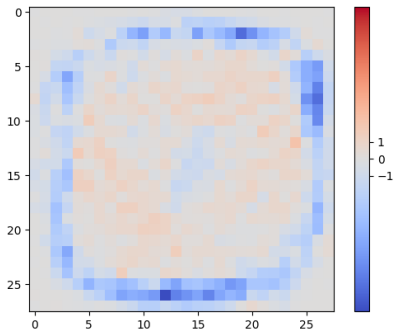
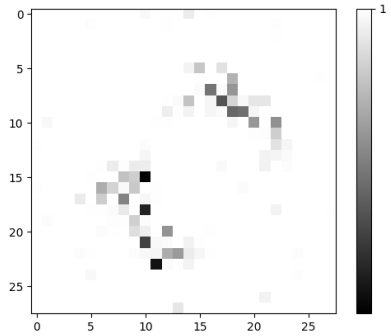


Figure: Features for a perceptron network



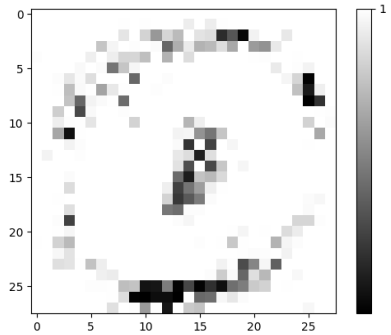


Figure: Logical Neural Network using an AND activation

Experimental Results: Interpretability - Hidden Layer

In this case, pictures represent an important feature for classifying an instance as a 1.

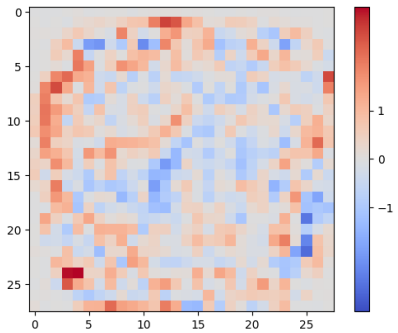
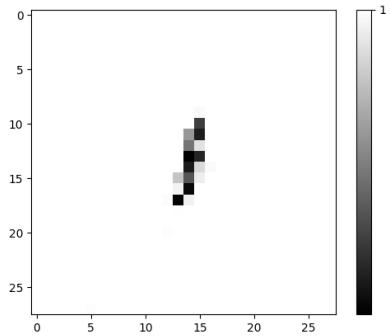


Figure: Features that positively contribute to the classification as a 1.



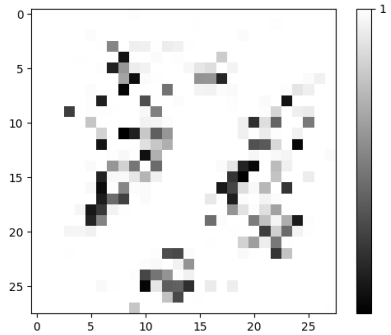


Figure: Features contributing to classification of a 1 in an AND $\rightarrow$ OR Model

Conclusion

Did we succeed in solving the problem? Well... Yes and No

- Logical Neural Networks are a promising alternative to Multi-Layer Perceptron Networks.
- Interpretability on MNIST was "better". However, this is difficult to establish.
- Can train shallow and deep networks with good accuracy.

Questions