

# Boolean Structured Deep Learning Network (BSnet)

Sing Kuang Tan  
Email: singkuangtan@gmail.com

December 26, 2022

## Abstract

In this paper, I am going to propose a new Boolean Structured Deep Learning Network (BSnet) based on the concept of monotone multi-layer Boolean algebra. I have shown that this network has achieved significant improvement in accuracy over an ordinary Relu Deep Learning Network.

## 1 Introduction

Last time when I was training face recognition model using deep learning, I hypothesized that the deep learning model is acting more like a nested Boolean operations rather than a general nested functions, where the actual output of each neuron is not important but it needs to know whether it is high or low. The high or low is input into a neuron which acts like a Boolean operator and output another high and low. The input is passed through a series of layers of Boolean operations and finally output a 0 or 1 whether it belongs to a class.

This Boolean representation makes the deep learning network more structured and have better training convergence. It also makes understanding of the deep network easier.

My Boolean Structured network has “Not” gates at the first layer, “And” and “Or” gates at the second layer. The next layer with “Not” gates and next layer after this is “And” and “Or” gates, and repeats this sequence for subsequent layers. This is similar to a monotone multi-layer Boolean algebra of layers where each layer contains only “And” and “Or” gates.

## 2 My Model

**Hypothesis: My BSnet has better performance than an ordinary Relu Deep Learning Network.**

I developed a Boolean Structured network (BSnet aka BullShit net, LOL Laugh Out Loud). Every complex layer of the input will concatenate the positive and negative vector of previous layer input. Then it will pass through a fully connected layer where all weights are constrained to be always positive or zero, and apply a Relu activation function to the output of fully connected layer. The concatenation of positive and negative vector represents the “Not” gates, and the fully connected layer represents the “And” and “Or” gates. It will go through 5 complex layers where there are 32 output neurons at each complex layer. Lastly the last complex layer will have sigmoid activation function to output the classification probabilities of each handwriting digit class. Input is  $28*28=784$  dimensions and output is 10 classes. My BSnet and ordinary network will each has the same 5 hidden layers with same number of neurons for fair comparison.

The figures of the BSnet and normal network are shown in Figure 1 and 2.

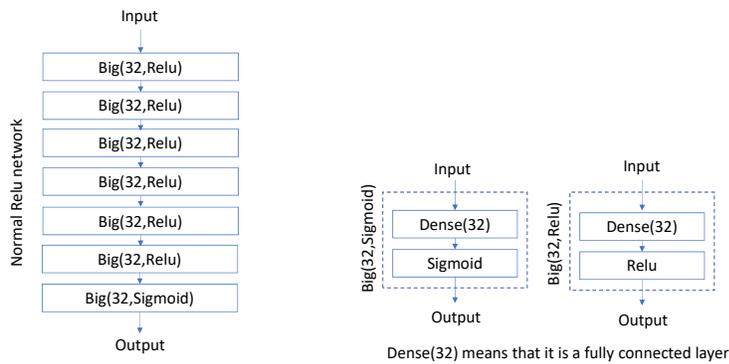


Figure 1: Network Diagram of Normal Relu Network

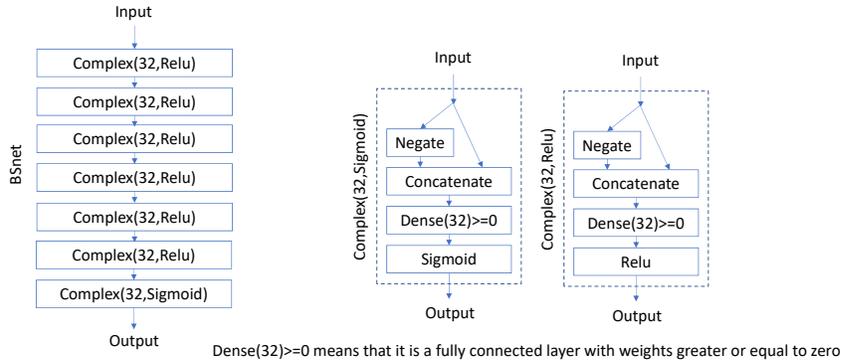


Figure 2: Network Diagram of BSnet

### 3 Experiment Results

To make the problem difficult, 70% of the input image pixels of the training set are masked (set to maximum value of 1), so that it is able to show that my BSnet is better. Otherwise, without the noise, both networks can easily achieved near 100% accuracies.

The figure below shows the validation and training curves of the two networks.

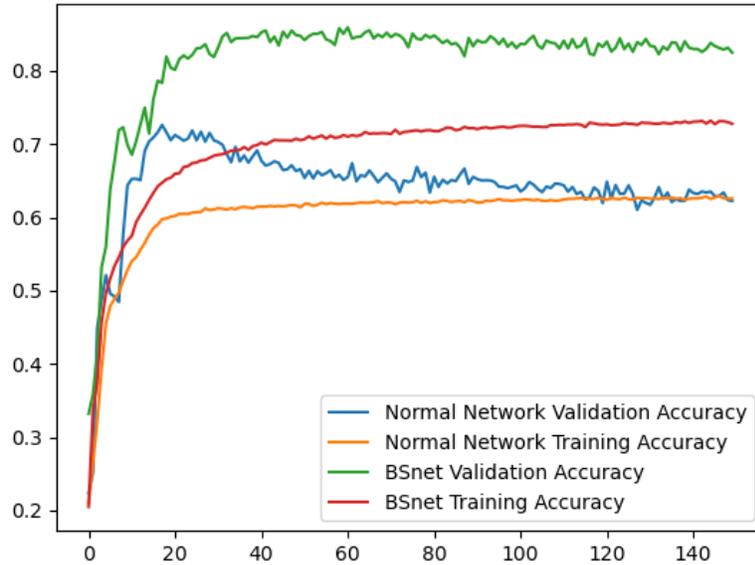


Figure 3: Validation and Training Accuracies of Normal Relu Network and BSnet

From the validation accuracy vs training epochs curves (see Figure 3), ordinary network shows signs of over fitting, with validation accuracy raises first then falls again. My BSnet will have higher accuracy than ordinary network.

LeNet has about 60000 parameters. My BSnet has 31552 parameters. You may think that making small changes (concatenate positive and negative vector of previous layer output, with weights constraint of greater or equal to zero) from the Normal network to BSnet result in small improvement. But for neurons with high dimensions input, the improvement is very obvious.

**I am going to make a claim here: Under certain conditions, my BSnet training optimization function is convex. In fully connected layer design (no convolution layers), my BSnet is still able to avoid overfitting and local minima's, and can perform better with less neurons and parameters than an ordinary network.**

Access my GitHub codes thru this link: <https://github.com/singkuangtan/BSnet>

## 4 Conclusion

I have developed a Boolean Structured Deep Learning Network for general classification problem in machine learning. The design in this network can be easily transferred to other type of tasks in machine learning such as long short-term

memory network, autoencoder, diffusion network, convolutional network, and recurrent network.

The idea is to make training algorithm or gradient descent of the deep learning network convex, so that training is easier and faster without many hyperparameters to tune. To get rid of convolution layers, dropout, weight decay, learning rate, batch normalization, other complicated and unnecessary designs so that the training process and the overall network structure become simple, easy to use and easy to design new networks for new problems.

The Boolean structure in the network is able to provide a theoretical model on how deep learning works, how it learns and how it can be used to model any general high dimensions function.