

On the Power of Threshold Circuits with Small Weights

Kai-Yeung Siu
Information Systems Laboratory
Stanford University
Stanford, CA 94305.

Jehoshua Bruck
IBM Research Division
Almaden Research Center
San Jose, CA 95120-6099

Abstract— Linear threshold elements (LTEs) are the basic processing elements in artificial neural networks. An LTE computes a function that is a sign of a weighted sum of the input variables. The weights are arbitrary integers; actually they can be very big integers—exponential in the number of input variables. However, in practice, it is very difficult to implement big weights. So the natural question that one can ask is whether there is an efficient way to simulate a network of LTEs with big weights by a network of LTEs with small weights. We prove the following results: 1) every LTE with big weights can be simulated by a depth-3, polynomial size network of LTEs with small weights, 2) every depth- d , polynomial size network of LTEs with big weights can be simulated by a depth- $(2d + 1)$, polynomial size network of LTEs with small weights. To prove these results, we use tools from harmonic analysis of Boolean functions. Our technique is quite general, it provides insights to some other problems. For example, we were able to improve the best known results on the depth of a network of threshold elements that computes the *COMPARISON*, *ADDITION* and *PRODUCT* of two n -bits numbers, and the *MAXIMUM* and the *SORTING* of n n -bit numbers.

Summary

The motivation for this work comes from the area of neural networks. A common model for neural network is the *threshold circuit*, a Boolean network in which every gate computes a *linear threshold function*. Many experimental results in this area have indicated that the magnitudes of the coefficients in the threshold elements grow very fast with the size of the inputs and therefore limit the practical use of the network. One natural question to ask is the following: How limited is the computational power of the network if we restrict ourselves to threshold elements

with only “small” growth in the coefficients? We answer this question by showing that we can trade-off an exponential growth with a polynomial growth in the magnitudes of coefficients by increasing the depth of the network by a factor of almost two and a polynomial growth in the size.

The fact that we can simulate a linear threshold function of exponentially large weights in a ‘constant’ number of layers of elements with ‘small’ weights follows from the results of Chandra et al. Their results showed that the sum of n n -bit numbers is computable in a constant number of layers of ‘counting’ gates, which in turn can be simulated by a constant number of layers of threshold elements with ‘small’ weights. However, it was not explicitly stated how many layers are needed in each step of their construction and direct application of their results would yield a constant such as 13. In this paper, we shall reduce the constant to 3 by giving a more ‘depth’-efficient algorithm and by using harmonic analysis of Boolean functions. We then generalize this result to higher depth circuits and show how to simulate a threshold circuit of depth- d and exponentially large weights in a depth- $(2d + 1)$ threshold circuit of ‘small’ weights.

As another application of harmonic analysis, we also show that the *COMPARISON* and *ADDITION* of two n -bit numbers is computable with only two layers of elements with ‘small’ weights, while it was only known to be computable in 3 layers. We also indicate how our ‘depth’-efficient algorithm can be applied to show that the product of two n -bit numbers can be computed in a depth-4 threshold circuit with polynomial number of gates. In addition, we show that the *MAXIMUM* and *SORTING* of n n -bit numbers can be computed in depth-3 and depth-4 threshold circuits with polynomial number of gates, respectively.