ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

Are Rosenblatt multilayer perceptrons more powerfull than sigmoidal multilayer perceptrons? From a counter example to a general result

J. Barahona da Fonseca

Department of Electrical Engineering, Faculty of Sciences and Technology,
New University of Lisbon
2829-516 Monte de Caparica, Portugal

Abstract. In the eighties the problem of the lack of an efficient algorithm to train multilayer Rosenblatt perceptrons was solved by sigmoidal neural networks and backpropagation. But should we still try to find an efficient algorithm to train multilayer hardlimit neuronal networks, a task known as a NP-Complete problem? In this work we show that this would not be a waste of time by means of a counter example where a two layer Rosenblatt perceptron with 21 neurons showed much more computational power than a sigmoidal feedforward two layer neural network with 300 neurons trained by backpropagation for the same classification problem. We show why the synthesis of logical functions with threshold gates or hardlimit perceptrons is an active research area in VLSI design and nanotechnology and we review some of the methods to synthesize logical functions with a multilayer hardlimit perceptron and we propose the search for an efficient method to synthesize any classification problem with analogical inputs with a two layer hardlimit perceptron as a near future objective. Nevertheless we recognize that with hardlimit multilayer perceptrons we cannot approximate continuous functions as we can easily do with multilayer sigmoidal neural networks, with multilayer hardlimit perceptrons we can only solve any classification problem, as we plan to demonstrate in a near future.

Keywords: Sigmoidal Neural Networks, Backpropagation, Multilayer Hardlimit Neural Networks, Efficient Training Algorithm for Multilayer Rosenblatt Perceptrons.

1 Introduction

Since the work of Minsky and Papert [1] in 1969 where they show that a hardlimit perceptron with one layer can classify only linearly separable problems and to classify problems linearly non separable it would be needed two or more layers, research in neural networks stagnated because there was no algorithm to train multilayer hardlimit perceptrons, only the Rosenblatt learning rule to train one layer perceptrons [2-3]. It was only in 1986 that the first answer to this open problem arose with multilayer sigmoidal neural networks and the Backpropagation algorithm [4]. Nevertheless research towards an algorithm to train multilayer hardlimit perceptrons or threshold gates to implement logical functions continued during the sixties [5-8] and was rediscovered by the VLSI and nanotechnology communities and some proposals recently arose [9-12] but until today nobody found an *efficient* algorithm to train multilayer hardlimit perceptrons to solve any classification problem, although there appeared some very inefficient proposals [13-15]. The synthesis of logical

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

functions with threshold began in forties [16] almost at the same time of the publication of the seminal work of McCulloch and Pitts [17] but it was only in the sixties that appeared the first proposals of algorithms to obtain the weights of multilayer threshold gates to implement logical functions. In this work we show that obtaining an efficient algorithm to design multilayer Rosenblatt perceptrons for classification problems it would not be a waste of time since in many cases, as in our counter example, multilayer hardlimit perceptrons exhibit a much greater computational power than multilayer sigmoidal networks trained by Backpropagation for the same classification task. Nevertheless surprisingly it has been shown that two layer sigmoidal networks are more powerful than two layer hardlimit networks for the class of Boolean functions [18].

2 The counter example: a five classes classification problem

Consider that we want to classify points belonging to four squares and we consider the complementary universe as a fifth class, a big square that contains the smaller four squares, as we show in figure 1.

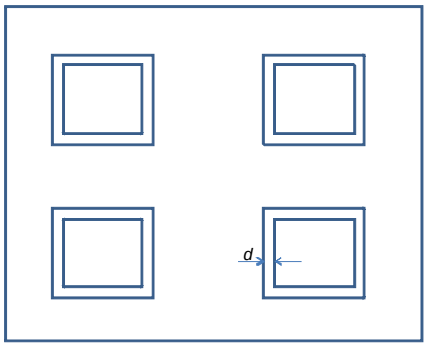

Fig. 1: Illustration of our classification problem. *d* is the *thickness* of the boundary between each of four classes and the universe considered as a fifth class defined by the big square that contains the small squares.

Since during the training of the sigmoidal neural network a point cannot belong simultaneously to one of the sides of the squares and to the complementary universe, we must consider a *boundary thickness, d,* and the points are only considered to belong to the complementary universe outside this boundary. With a two layer feedforward neural networks with N sigmoidal neurons in the first layer and four linear neurons in the second layer, using the output codification $[1\ 0\ 0\ 0]^T$ for the first square, $[0\ 1\ 0\ 0]^T$ for the second square and so on, and with a *test set* of 10M points we verified that as we reduce the thickness d we must increase the number of hidden neurons N to obtain good classification results. For a complementary universe with side 6, squares with side 0.4 and d=0.1 we needed N=300 hidden sigmoidal neurons to obtain good classification results, greater than 99%, using the

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

Levenberg-Marquardt backpropagation, and we needed N=250 hidden sigmoidal neurons using the variant of Levenberg- Marquardt backpropagation with Bayesian Regularization [19] to attain the same classification results. When we decrease d the classification performance degraded and it would be necessary to increase even more the number of hidden neurons. But a square can be seen as the intersection of four semi planes, each semi plane being defined by a Rosenblett perceptron in the first layer and the intersection of semi planes being implemented by a fifth Rosenblatt perceptron in the second layer that makes the logical AND of the output of the four hardlimit perceptrons of the first layer. To classify the complementary universe with any dimension we only need one more hardlimit perceptron that implements the logical NOR of the classifications of the four groups:

$H(-c_1 - c_2 - c_3 - c_4)$

where the c_i are the outputs of the hardlimit perceptrons that classify the points belonging to each of the four squares and H is defined by (1)

$$H(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{1}$$

As an example, consider a square defined by the following semi planes, considering that
$x_1 < x_2$ and $y_1 < y_2$:

$x \geq x_1$, which is classified by the hardlimit perceptron $o_1=H(x - x_1)$

$x \leq x_2$, which is classified by the hardlimit perceptron $o_2=H(- x + x_2)$

$y \geq y_1$, which is classified by the hardlimit perceptron $o_3=H(y - y_1)$

$y \leq y_2$, which is classified by the hardlimit perceptron $o_4=H(- y + y_2)$

and the logical AND of the four o_i is implemented by the following fifth hardlimit perceptron that classifies if any point (x, y) belongs to one of four squares defined by the vertices (x_1, y_1), (x_1, y_2), (x_2, y_1) and (x_2, y_2):

$c_1=H(o_1 + o_2 + o_3 + o_4 - 4)$

To classify the complementary universe, u, we just make the logical NOR of the four c_i that classify the four squares with the following hardlimit perceptron

$u= H(-c_1 - c_2 - c_3 - c_4)$

This way with only 21 hardlimit perceptrons we can classify any four squares with a boundary with null thickness, d=0, and an infinite complementary universe as the fifth

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

class. On the contrary with the sigmoidal network as we increase the complementary universe and decrease the boundary thickness the number of hidden neurons must increase exponentially. This counter example shows that although it does not exist an efficient learning algorithm to solve classification problems for multilayer Rosenblatt perceptrons, these multilayer hardlimit perceptrons are in some cases much more computationally powerful than sigmoidal feedforward neural networks for the same classification problem.

3 A two layer rosenblatt perceptron can compute any logical function

Next we will show the known result that a two layer Rosenblatt perceptron is capable to compute any logical function by an original constructivist argument. It is this result that makes the VLSI and nanotechnology communities interested in implementing logical functions with threshold gates or hardlimit perceptrons because their implementation with threshold gates is in many cases much simpler than with digital gates and it can be saved lots of semiconductor space.
Consider a logical function of four variables in the first normal form given by (2).

$$F(x_1, x_2, x_3, x_4)=m_1 + m_7 + m_{14} + m_{15} \quad (2)$$

Since any minterm is a linearly separable function, it is the logical AND of positive and negated variables, the hardlimit perceptrons of the first layer will implement the minterms and in the second layer a hardlimit perceptron will implement the logical OR of these minterms. So we could train each hardlimit perceptron with the perceptron learning rule to obtain the right weights. Nevertheless, due to the simplicity of the minterms we will present a method to assign the right weights. The weights associated to the positive variables are set to 1, the weights associated to negated variables are set to -1 and the bias is set to $-N$, N being the number of positive variables. For example the minterm $m_7 = x_1 x_2 x_3 \bar{x}_4$ is implemented by a hardlimit perceptron with weights given by $w_1=w_2=w_3=1$, $w_4=-1$, and bias given by $b=-3$.

Finally the OR of the minterms is implemented by a hardlimt perceptron in the second layer with weights set to 1 and bias set to -1. This function can be minimized using Karnaugh maps resulting in the simplified expression

$$F'(x_1, x_2, x_3, x_4)=x_1 x_2 x_3 + x_2 x_3 x_4 + x_1 x_2 x_3 x_4 \quad (3)$$

which can be implemented by four hardlimit perceptrons. It can be shown that any association of two of these terms is a linearly non separable function so it cannot be further minimized. Nevertheless this is not a general result.
We can generalize this constructivist design algorithm to any logical function with N minterms, being each minterm implemented by a perceptron and a $N+1th$ perceptron that implements the OR of the N minterms. So we found a constructive methodology to build any logical function with N minterms with a two layer hardlimit perceptron

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

with N+1 neurons, which proves that any logical function can be implemented by a two layer hardlimit perceptron, as we wanted to show. In most cases the terms of the minimized function cannot be associated, i.e. from their association there results a linearly non separable logical function, and so the implementation of the minimized function corresponds to the minimum number of threshold gates.
From this result we can design a new approach to digital design where the logical gates are replaced by perceptrons implemented by a single high gain transistor and some resistances. This could save a lot of silicon space as in the case of the implementation of a N bits comparator that is implemented by a single perceptron but whose implementation with logical gates implies a number of gates that increases exponentially with N.

4 Conclusions and Future Work

We showed by means of a counter example that in many cases multilayer hardlimit perceptrons have more computational power than multilayer sigmoidal neural networks to solve a given classification problem. As a corollary of the demonstration of the universality of a two layer hardlimit perceptron we showed how to implement logical functions with operational amplifiers which in many cases have a great advantage in terms of the number of transistors needed to implement them over the classical digital design as it is the case of the majority function of N bits and the N bits comparator, the implementation of which with digital design would imply a number of logical gates that increases exponentially with N. In a near future we will work towards finding an efficient algorithm to train multilayer hardlimit perceptrons to solve classification tasks, a problem that is shown to be NP-Complete.

References

[1] M. Minsky and S. Papert. *Perceptrons*, MIT Press, Washington DC, 1969.

[2] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, *Psychological Review*, 65: 386-408, 1958.

[3] F. Rosenblatt, *Principles of Neurodynamics*, Spartan Press, Washington DC, 1961.

[4] D. E.Rumelhart, G. E. Hinton and R. J. Williams, Learning representations by back-propagating errors, *Nature*, 323:533-536, 1986.

[5] S.H Cameron, The generation of minimal threshold nets by an integer program, *IEEE Transactions on Electronic Computers*, EC-13: 3: 299-302, 1964.

[6] Hopcroft, J. E. and R. L. Mattson, Synthesis of minimal threshold logic networks, *IEEE Transactions on Electronic Computers*, EC-14: 4: 552- 560, 1965.

[7] P. K. S. Roy, Synthesis of symmetric switching functions using threshold logic elements, *IEEE Transactions on Electronic Computers*, EC-16: 3: 359-364, 1967.

[8] S. Ghosh, D. Basu, and A. K. Choudhury, Multigate synthesis of general boolean functions by threshold logic elements, *IEEE Transactions on Computers*, Vol. C-18, *n*. 5, pp. 451-456, 1969.

[9] A. L. Oliveira, and A. Sangiovanni-Vincentelli, LSAT: An algorithm for the synthesis of two-level

threshold gate networks. In: Proceedings of IEEE Int. Conf. On CAD, ICCAD'91, pp. 130-133, 1991.

[10] R. Zhang, P. Gupta, L. Zhong, and N. K. Jha, Threshold network synthesis and optimization an application to nanotechnologies, *IEEETransactions on Computer-Aided Design of Integrated Circuits and Systems,* 24: 1: 107-118, 2005.

[11] K. Aoyama, Operating margin-oriented design methods for threshold element-based reconfigurable logic circuits realizing any Symmetric Function, *Journal of VLSI Signal Processing*, 38: 157–171, 2004.

[12] M. J. Avedilo and J. M. Quintana, A threshold logic synthesis tool for RTD circuits. In: Proceedings of the EUROMICRO Systems on Digital System Design, DSD'04, pp. 624-627, 2004.

[13] E. M. Corwin, A. M. Logar and W. J. B. Oldham, An iterative method for training multilayer networks with threshold functions, *IEEE Transactions on Neural Networks*, 5:3:,507-508, 1994.

[14] A. Yamamoto, and T. Saito, An improved expand-and-truncate learning, in *IEEE International Conference on Neural Networks*, 2: 1111-1116, 1997.

[15] M. Tajine, and D. Elizondo, The recursive deterministic perceptron neural network, *Neural Networks*, 1571-1588, 1998.

[16] V. Beiu, Threshold logic implementations: the early days, In *Proceedings of 46th IEEE Midwest Symposium on Circuits and Systems,* 3: 1379-1383, 2003.

[17] W. S. McCulloch, and W. H. Pitts, A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, 5:115-133, 1943.

[18] Maass, W., Schnitger, G. and Sontag, E. D., A Comparison of the Computational Power of Sigmoid and Boolean Threshold Circuits, In V. P. Roychowdhury, Siu K. Y., and Orlitsky A., editors, *Theoretical Advances in Neural Computation and Learning,* 127-151, Kluwer Academic Publishers, 1994.

[19] K. Tin-Yau, and Y. Dit-Yan, Baysean regularization in constructive neural networks, In C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen and B. Sendhoff (Eds.), *Lecture Notes in Computer Science*, 1112: 557-562, 1996.