ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 341-346

Multi-Sigmoidal Units and Neural Networks

John A. Drakopoulos

Stanford University
Department of Computer Science
Knowledge Systems Laboratory
701 Welch Road, Palo Alto, CA 94304-0106

Abstract. *Multi-sigmoidal units*, which are a generalization of ordinary sigmoidal units used in neural networks, are introduced here. *Multi-sigmoidal neural networks* (i.e. neural networks with multi-sigmoidal units) share the same function representation and approximation capabilities with ordinary sigmoidal neural networks. In addition, it is proved here that if X is a finite and discrete subset of real numbers then any function $f : X^n \rightarrow \{0, \ldots, C-1\}^\lambda$ can be represented by a two-layer network (i.e. no hidden units) that has $\lambda\lceil \log_2 C \rceil$ multi-sigmoidal output units. The above result indicates that there is a trade-off in capturing interactions among inputs and representing functions either via hidden units or via non-monotonic unit activation functions.

1. Introduction

Perceptrons have been severely criticized for their inability to represent simple boolean functions (such as exclusive-OR) [12]. As it has been shown in the literature [16], the above limitation is mainly due to the fact that perceptrons are two-layer networks (i.e. they have no hidden layers). Three-layer neural networks with arbitrary activation functions (especially in the upper layer) can represent any continuous function [8, 17, 4]. On the other hand, given a particular activation function ψ, three-layer neural networks can uniformly approximate any continuous function, if and only if ψ is non-polynomial [10, 6]. Furthermore, if ψ is a squashing function then three-layer neural networks can uniformly approximate and Borel measurable function [7]. Based on the later results, in many neural network applications, simple monotonic activation function (such as sigmoidal functions) have been used [16, 5, 11].

In this paper, it is proved that some of the computational limitations of two-layer networks with monotonic activation functions are due to the monotonicity of their activation functions. In our study, units have activation functions that consist of a collection of simple sigmoidal functions arranged along a dimension so that they constitute a piecewise sigmoidal or *multi-sigmoidal* function. Those units are called *multi-sigmoidal units.*

It is proved that, if X is a discrete and finite subset of real numbers then any function $f : X^n \rightarrow \{0, 1\}$ can be represented by a single multi-sigmoidal unit.

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 341-346

As a consequence, any function $f : X^n \to \{0, \ldots, C-1\}^\lambda$ can be represented by a two-layer network (i.e. no hidden units) that has $\lambda\lceil \log_2 C \rceil$ multi-sigmoidal output units.

The above results indicate that there is a trade-off in capturing interactions among inputs and representing or approximating functions using either hidden units or non-monotonic unit activation functions.

Non-monotonic activation functions are not uncommon in neural network literature [11, 15, 14, 1]. Empirical comparisons of sigmoids and radial basis functions appear in [18, 13, 9] while empirical comparisons of sigmoids, polynomials, rational functions, and flexible Fourier series appear in [14]. All those studies provide some empirical evidence that non-monotonic unit activation functions could aid representation and approximation of functions by neural networks.

In the next section, formal definitions are followed by two theorems regarding the representational capabilities of two-layer multi-sigmoidal neural networks. In section 3, an example of a multi-sigmoidal unit is presented, and the representational capabilities of multilayer multi-sigmoidal networks are discussed.

2. Multi-Sigmoidal Units and Neural Networks

Sigmoidal functions are formally defined and studied in [2]. Here, a *basic unit sigmoidal function* S is used and its affine class A_S is defined as follows:

$$A_S = \{S_{a,c}/a, c \in \mathcal{R}, a \neq 0 \quad and \quad \forall x \quad S_{a,c}(x) = S(ax - c)\}$$

where $\mathcal{R}$ is the set of real numbers. Intuitively, a determines the steepness and monotonicity of $S_{a,c}$ while the ratio c/a determines the position of the center of symmetry of $S_{a,c}$. For example, S could be chosen to be the logistic function.

Now, define $[\phi]$ to be equal to 1 if the condition ϕ is true and equal to zero if the conditions ϕ is false. Then

Definition 1 *A function f is called a multi-sigmoidal function iff*

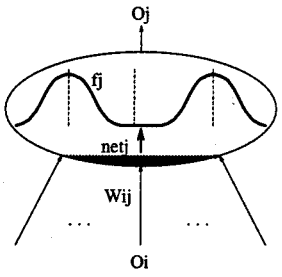
$$\exists k, a_1, \ldots, a_k, c_1, \ldots, c_k \qquad \forall i \;\; a_{i-1}a_i < 0 \quad and$$
$$\forall x \;\; f(x) = \sum_{i=1}^{k} S_{a_i,c_i}(x)[b_{i-1} < x \leq b_i]$$

where S is a sigmoidal function, $b_0 = -\infty$, $b_k = +\infty$, and $b_i = \frac{c_{i+1}-c_i}{a_{i+1}-a_i}$, $i = 1, 2, \ldots, k-1$. In that case we write $f = (S_{a_1,c_1}, b_1, S_{a_2,c_2}, b_2, \ldots, b_{k-1}, S_{a_k,c_k})$.

We call a_i, c_i $(i = 1, \ldots, k)$ the *affine coefficients* and b_i $(i = 1, \ldots, k)$ the *barrier locations* of the *ms* function f. Note that the conditions upon b_0 and b_k are for notational convenience while the condition upon b_i results in a function f that is continuous[1] at b_i, for $i = 1, 2, \ldots, k-1$.

[1]This condition is not only sufficient but also necessary when S is invertible. The later is equivalent to the generator of S being zero in a set of points that have zero total length which, in turn, is equivalent to S being strictly monotonic.

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 341-346



$$o_j = f_j(net_j)$$

$$f_j = (S_{a_{j1},c_{j1}}, b_{j1}, \ldots, b_{j,k-1}, S_{a_{jk},c_{jk}}), \text{ for some } k \in \mathcal{N}$$

$$net_j = \sum_i o_i w_{ij}$$

Figure 1: A Typical MS Unit. net_j is the net input to node j, o_j is its output, and f_j is its activation function.

Now, a multi-sigmoidal unit (MS unit) is a unit with a multi-sigmoidal activation function while a multi-sigmoidal neural network (MSNN) is a neural network whose units are MS units. A typical MS unit and its functionality are shown in figure 1.

Now, we can state and prove the following theorem:

Theorem 1 *Let $X \subset \mathcal{R}$ be a finite and discrete set and S be a unit sigmoidal function with range R_S (i.e. $(0,1) \subseteq R_S \subseteq [0,1]$). Then*

$$\forall n \geq 1 \quad \forall f : X^n \to \{0,1\} \quad \exists W \in \mathcal{R}^n \quad \forall \epsilon > 0$$
$$\exists k, a_1, \ldots, a_k, c_1, \ldots, c_k, b_1, \ldots, b_{k-1} \quad \forall u \in X^n \quad |f(u) - g(W \cdot u)| \leq \epsilon$$

where $g = (S_{a_1,c_1}, b_1, S_{a_2,c_2}, b_2, \ldots, b_{k-1}, S_{a_k,c_k})$. Furthermore, if $R_S = [0,1]$ then ϵ could be chosen to be zero (i.e. $f = g$) in the above inequality.

Proof
Let $m = \min\{|x-y| \ / \ x, y \in X, x \neq y\}$, $M = \max\{|x-y| \ / \ x, y \in X\}$, $B = \frac{M}{m} + 1$, and $W = [B^0, B^1, \ldots, B^{n-1}]^T$. Then the function $h(u) = W \cdot u$, $u \in X^n$ can be proven to be $1-1$ in X^n. Thus we can write X^n as $\{u_1, u_2, \ldots, u_N\}$ so that $h(u_i) < h(u_{i+1})$ $(i = 1, 2, \ldots, N-1)$, where N is the finite cardinal of X^n.

Now we can construct g. To this purpose, let $u_{i_1}, u_{i_2}, \ldots, u_{i_k}$ denote the elements of X^n where it is $f(u_{i_j}) \neq f(u_{i_j+1})$ $(j = 1, 2, \ldots, k-1)$. Then define $g = (S_{a_1,c_1}, b_1, S_{a_2,c_2}, b_2, \ldots, b_{k-1}, S_{a_k,c_k})$ where $|S_{a_j,c_j}(h(u_{i_j})) - f(u_{i_j})| = |S_{a_j,c_j}(h(u_{i_j+1})) - f(u_{i_j+1})| = \epsilon$. Obviously, since $\{f(u_{i_j}), f(u_{i_j+1})\} = \{0,1\}$, ϵ could be chosen to be zero in the above equations, if $R_S = [0,1]$. In either case, those equations can define solution sets for a_j, c_j $(j = 1, 2, \ldots, k)$. Any element in those solution sets will be an acceptable assignment of values to the affine coefficients a_j, c_j $(j = 1, 2, \ldots, k)$. The barrier locations are now defined as $b_j = \frac{c_{j+1} - c_j}{a_{j+1} - a_j}$, $j = 1, 2, \ldots, k-1$.

Now, $\forall u \in X^n \quad \exists j \quad h(u_{i_j}) \leq h(u) \leq h(u_{i_j+1})$. Since, S is monotonic in $[h(u_{i_j}), h(u_{i_j+1})]$, and f is constant over all $u \in X^n$ that map (through h) into $[h(u_{i_j}), h(u_{i_j+1})]$, the required inequality $|f(u) - g(W \cdot u)| \leq \epsilon$ holds. Q.E.D.

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 341-346

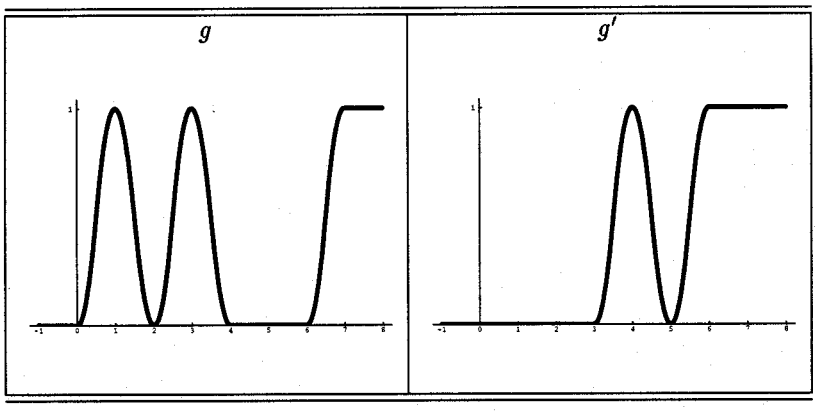


Figure 2: Multi-sigmoidal functions for $f(x, y, z) = (x \wedge y) \vee (x \wedge \bar{z})$.

The above theorem states that one multi-sigmoidal unit is necessary in order to represent any function $f : X^n \rightarrow \{0, 1\}$. As a direct consequence, λ multi-sigmoidal units are necessary to represent any function $f : X^n \rightarrow \{0, 1\}^\lambda$. Furthermore, since $\lceil \log_2 C \rceil$ bits are necessary to represent any number in $\{0, 1, \ldots, C-1\}$ we have the following theorem:

Theorem 2 *If $X \subset \mathcal{R}$ is discrete and finite then any function $f : X^n \rightarrow \{0, 1, \ldots, C-1\}^\lambda$ can be represented by a two-layer multi-sigmoidal network having $\lambda \lceil \log_2 C \rceil$ multi-sigmoidal units.*

3. Examples and Discussion

Consider the boolean function $f(x, y, z) = (x \wedge y) \vee (x \wedge \bar{z})$. Now, it is $X = Y = \{0, 1\}$. Thus (see proof of theorem 1), it would be $M = m = 1$, $B = 2$, and so it would be $h(x, y, z) = x + 2y + 4z$. This results to a multi-sigmoidal g function as shown in figure 2. However, if we define $h(x, y, z) = 4x + 2y + z$ we get the multi-sigmoidal function g' that has fewer sigmoidal functions than g. In adition, for the n-parity problem, it can be proved by induction that a weight vector as the one constructed in theorem 1 will require $\lfloor \frac{2^{n+1}-1}{3} \rfloor$ sigmoidal functions. On the other hand, setting all weights equal to a constant (say 1) will result in a multi-sigmoidal unit that can represent n-parity using only n sigmoidal components.

In general, for any given problem, the weight vector W may affect the number of sigmoidal components of g. Finding a weight vector that would minimize the number of sigmoidal functions of an MS unit is an open problem and should be a subject of further research.

However, two-layer multi-sigmoidal networks cannot represent or approximate arbitrary functions, by any $\epsilon > 0$. For example, the two-spiral problem

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 341-346

cannot be modellled by a single multi-sigmoidal unit. On the other hand, multi-sigmoidal units are generalizations of ordinary sigmoidal units since every sigmoidal unit is a multi-sigmoidal unit with only one sigmoidal component. Thus, the functions that are representable by ordinary sigmoidal neural networks are representable by multi-sigmoidal networks, too. As a consequence, three-layer multi-sigmoidal networks can uniformly approximate any (Borel) measurable function [7].

Unfortunately, there has not been discovered a theorem such as theorem 1 that would provide a weight vectors and multi-sigmoidal functions for multilayer multi-sigmoidal neural networks. However, a heuristic and greedy algorithm (steepest descent) that trains multilayer multi-sigmoidal networks and dynamically adapts the sigmoidal components on each unit is the multi-sigmoidal back propagation algorithm that is presented in [3].

Finally, it must be noted that two layer multi-sigmoidal neural networks can represent strictly more functions than two layer ordinary sigmoidal neural networks can (e.g. consider the XOR function). This is due to the fact that non-monotonic unit activation functions can help to represent functions with either fewer units or fewer layers.

4. Conclusion

Multi-sigmoidal units reveal an existing trade-off in representing functions using hidden units or non-monotonic unit activation functions. Although the theorems proved here shed some light into this trade-off much remains to be studied and analyzed. Furthermore, both mechanisms (i.e. hidden units and non-monotonicity) could be combined in order to optimize one's network.

References

[1] M.R.W. Dawson and D.P. Schopflocher. Modifying the generalized delta rule to train networks of non-monotonic processors for pattern classification. *Connection Science*, 4(1):19–31, 1992.

[2] J.A. Drakopoulos. Sigmoidal theory. Technical Report KSL–94–41, Knowledge Systems Laboratory, Stanford University. To appear in the *International Journal of Fuzzy Sets and Systems*, 1994.

[3] J.A. Drakopoulos. Multi-sigmoidal neural networks and back-propagation. Submitted to *Fourth International Conference on Artificial Neural Networks*, 1995.

[4] R. Hecht-Nielsen. Kolmogorov's mapping neural network existence theorem. In *Proceedings of the International Conference on Neural Networks, III*, pages 11–13, New York, 1987. IEEE Press.

[5] G.E. Hinton. Connectionist learning procedures. *Artificial Intelligence*, 40:185–234, 1989.

[6] K. Hornik. Some new results on neural network approximation. *Neural Networks*, 6:1069–1072, 1993.

[7] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[8] A.N. Kolmogorov. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Doclady Academy Nauk SSSR*, 114:953–956, 1957.

[9] A.S. Lapedes and R.M. Farber. Nonlinear signal processing using neural networks: prediction and system modeling. Technical Report LA-UR–87–2662, Los Alamos National Laboratory, 1987.

[10] M. Leshno, V.Y. Lin, A. Pinkus, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6:861–867, 1993.

[11] R.P. Lippmann. Pattern classification using neural networks. *IEEE Communications Magazine*, November, 27(11):47–64, 1989.

[12] M.L. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry.* MIT Press, Cambridge, MA, expanded edition, 1988.

[13] J. Moody and C.J. Darken. Fast learning in networks of locally tuned processing units. *Neural Computation*, 1:281–294, 1989.

[14] J. Moody and N. Yarvin. Networks with learned unit response functions. In J.E. Moody, S.J. Hanson, and R.P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 1048–1055. Morgan Kaufmann, San Mateo, CA, 1992.

[15] T. Poggio and F. Girosi. A theory of networks for approximation and learning. Technical Report No. 1140, MIT AI Memo Lab, 1989.

[16] D.E. Rumelhart, G.E. Hinton, and R.J.Williams. Learning internal representations by error propagation. In D.E. Rumelhart, J.L. McClelland, and the PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, I: Foundations*, pages 318–363. MIT Press, Cambridge, MA, 1986.

[17] D.A. Sprecher. On the structure of continuous functions of several variables. *Transactions American Mathematical Society*, 115(3):340–355, 1965.

[18] S.A. Weigend, B.A. Hubermann, and D.E. Rumelhart. Predicting the future: A connectionist approach. *International Journal of Neural Systems*, 1:193–209, 1990.