LA-UR- 9 8 - 1 6 0 9

CONF-980942--

Title: ON ANALOG IMPLEMENTATIONS OF DISCRETE
NEURAL NETWORKS

Author(s): Valeriu Beiu, NIS-1
Kurt R. Moore, NIS-1

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Los Alamos
NATIONAL LABORATORY

DISCLAIMER

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

ON ANALOG IMPLEMENTATION OF DISCRETE NEURAL NETWORKS

V. BEIU [a], K.R. MOORE

Los Alamos National Laboratory
Division Space and Atmospheric Sciences NIS-1, MS D466
Los Alamos, New Mexico 87545, USA
E-mail: {beiu, krmoore}@lanl.gov

Abstract—The paper will show that in order to obtain minimum *size* neural networks (*i.e.*, *size*-optimal) for implementing any Boolean function, the nonlinear activation function of the neurons has to be the identity function. We shall shortly present many results dealing with the approximation capabilities of neural networks, and detail several bounds on the *size* of threshold gate circuits. Based on a constructive solution for Kolmogorov's superpositions we will show that implementing Boolean functions can be done using neurons having an identity nonlinear function. It follows that *size*-optimal solutions can be obtained only using analog circuitry. Conclusions, and several comments on the required precision are ending the paper.
Keywords—neural networks, Kolmogorov's superimpositions, threshold gate circuits, analog circuits, size, precision.

1 Introduction

In this paper a *network* is an acyclic graph having several input nodes, and some (at least one) output nodes. If a synaptic *weight* is associated with each edge, and each node computes the weighted sum of its inputs to which a nonlinear activation function is then applied (*artificial neuron*): $f(x) = f(x_1, ..., x_\Delta) = \sigma\left(\sum_{i=1}^{\Delta} w_i x_i + \theta\right)$, the network is a *neural network* (NN), with the synaptic *weights* $w_i \in \mathbb{R}$, $\theta \in \mathbb{R}$ known as the *threshold*, Δ being the *fan-in*, and σ a non-linear activation function. Because the underlying graph is acyclic, the network does not have feedback connections, and can be layered. That is why such a network is also known as a *multilayer feedforward neural network*, and is commonly characterised by two cost functions: its *depth* (*i.e.*, number of layers), and its *size* (*i.e.*, number of neurons).

The paper starts by presenting known results dealing with the approximation capabilities of NNs, and details several bounds on the *size* of *threshold gate circuits* (TGCs). Based on a constructive solution for Kolmogorov's superpositions we will show that in order to obtain minimum *size* NNs (*i.e.*, *size*-optimal) for implementing any *Boolean function* (BF), the nonlinear activation function of the neurons has to be the identity function. Hence, *size*-optimal hardware implementations of discrete NNs (*i.e.*, implementing BFs) can be obtained only in analog circuitry. Conclusions, and several comments on the required precision are ending the paper.

---

[a] On leave of absence from the "Politehnica" University of Bucharest, Computer Science Department, Spl. Independenţei 313, RO-77206 Bucharest, România.

2 Previous Results

NNs have been experimentally shown to be quite effective in many applications (see *Applications of Neural Networks*[1], together with *Part F: Applications of Neural Computation*[2] and *Part G: Neural Networks in Practice: Case Studies*[2]). This success has led researchers to undertake a rigorous analysis of the mathematical properties that enable them to perform so well, and has generated two directions of research: (*i*) to find existence/constructive proofs for what is now known as the "*universal approximation problem;*" (*ii*) to find tight bounds on the *size* needed by the approximation problem (or some particular cases). The paper will focus on both aspects, for the particular case when the functions to be implemented are BFs.

2.1 Neural Networks as Universal Approximators ($f: \mathbb{R}^n \to \mathbb{R}$)

The first line of research on the approximation capabilities of NNs[3-6] was started in 1987 by Hecht-Nielsen[7] and Lippmann[8] who, together with LeCun,[9] were probably the first to recognise that the specific format of the form[10,11]:

$$f(x_1, \ldots, x_n) = \sum_{q=1}^{2n+1} \left\{ \Phi_q \left[\sum_{p=1}^{n} \alpha_p \, \psi \, (x_p + qa) \right] \right\} \tag{1}$$

of Kolmogorov's superpositions[12]:

$$f(x_1, \ldots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \, (y_q) \tag{2}$$

can be interpreted as a NN with one hidden layer. This gave an existence proof of the approximation properties of NNs. The first nonconstructive proof has been given the next year by Cybenko[13,14] using a continuous activation function and was independently presented by Irie and Miyake.[15] Thus, the fact that NNs are computationally universal—with more or less restrictive conditions—when modifiable connections are allowed, was established. Different enhancements have been later presented in the literature[16,17(Chp. 1)]:

- Funahashi[18] proved the same result in a more constructive way and also refined the use of Kolmogorov's theorem,[7] giving an approximation result for two-hidden-layer NNs;
- Hornik *et al.*[19] showed that the continuity requirement for the output function can be partly removed;
- Hornik *et al.*[20] also proved that a NN can approximate simultaneously a function and its derivative;
- Park and Sandberg[21,22] used radial basis functions in the hidden layer, and gave an 'almost' constructive proof;
- Hornik[23] showed that the continuity requirement can be completely removed, the activation function having to be 'bounded and nonconstant';
- Geva and Sitte[24] proved that four-layered NNs with sigmoid activation function are universal approximators;

- Kůrková[25] has demonstrated the existence of approximate superposition representations, *i.e.* ψ and Φ_q can be approximated with functions of the form $\sum a_r \sigma (b_r x + c_r)$, where σ is an arbitrary activation sigmoidal function;
- Mhaskar and Micchelli[26,27] approach was based on the Fourier series of the function, by truncating the infinite sum to a finite set, and rewriting e^{ikx} in terms of the activation function (which has to be periodic);
- Koiran[28] presented a new proof on the line of Funahashi's,[18] but more general in that it allows the use of units with 'piecewise continuous' activation functions; these include the important case of *threshold gates* (TGs);
- Leshno *et al.*[29] relaxed the condition for the activation function to 'locally bounded piecewise continuous' (*i.e.*, if and only if the activation function is not a polynomial), thus embedding as special cases almost all the activation functions that have been previously reported in the literature;
- Hornik[30] added to these results by proving that: (*i*) if the activation function is locally Riemann integrable and nonpolynomial, the *weights* and the *thresholds* can be constrained to arbitrarily small sets; and (*ii*) if the activation function is locally analytic, a single universal *threshold* will do;
- Funahashi and Nakamura[6] showed that the universal approximation theorem also holds for trajectories of patterns;
- Sprecher[31] has demonstrated that there are universal hidden layers that are independent of the number of input variables n;
- Barron[32] described spaces of functions that can be approximated by Jones' algorithm[33] using functions computed by single-hidden-layer network of TGs;

All these results—with the partial exception of[21,22,25,28,32]—were obtained *"provided that sufficiently many hidden units are available"* (*i.e.*, no claims on the *size* minimality were made). More constructive solutions have been obtained in very small *depth* later,[34–36] but their *size* grows fast with respect to the number of dimensions n and/or examples m, or with the required precision. *Recently, an explicit numerical algorithm for superpositions has been detailed.*[37–39]

2.2 Bounds on the Size of Threshold Gate Circuits

The other line of research was to find the smallest *size* NN which can realise an arbitrary function given a set of m vectors (*i.e.*, examples) from $\mathbb{R}^n$. Most of the results have been obtained for TGs[40]. Probably the first lower bound on the *size* of a *threshold gate circuit* (TGC) for almost all n-ary BFs (*i.e.*, $f: \mathbb{B}^n \to \mathbb{B}$) was[41]:

$$size \geq 2 (2^n/n)^{1/2}, \tag{3}$$

while later a very tight upper bound has been proven[42] in *depth* $= 4$:

$$size \leq 2\,(2^n/n)^{1/2} \times \{1 + \Omega\,[(2^n/n)^{1/2}]\}. \tag{4}$$

A similar existence lower bound for arbitrary BFs[43] is $\Omega\,(2^{n/3})$, while Roychowdhury *et al.*[44] details lower bounds for particular BFs.

For classification problems ($f: \mathbb{R}^n \to \mathbb{B}^k$), one of the first results was that a NN of *depth* = 3 and *size* = m–1 could compute an arbitrary dichotomy (*i.e.*, $k = 1$). The main improvements have been:

- Baum[45] presented a TGC with one hidden layer having $\lceil m/n \rceil$ neurons capable of realising an arbitrary dichotomy on a set of m points *in general position* in $\mathbb{R}^n$; if the points are on the corners of the n-dimensional hypercube (*i.e.*, $f: \mathbb{B}^n \to \mathbb{B}$), $m - 1$ nodes are still needed;
- a slightly tighter bound of only $\lceil 1 + (m-2)/n \rceil$ neurons in the hidden layer for realising an arbitrary dichotomy on a set of m points which satisfy a more relaxed topological assumption was proven later;[46] also, the $m - 1$ nodes condition was shown to be the least upper bound needed;
- Arai[47] showed that $m - 1$ hidden neurons are necessary for arbitrary separability (any mapping between input and output for the case of binary-valued units), but improved the bound for the dichotomy problem to $m/3$ (without any condition on the inputs);
- Beiu and De Pauw[48] detailed tight existence lower and upper bounds for arbitrary BFs: $(m/n)\,(1/2 + 2/\log n) < size < 0.72\,(m/n)\,(1/2 + 2/\log n)\,\log n$; they have been obtained by estimating the entropy of the data-set.

Several other existence lower bounds for arbitrary dichotomy are as follows[49,50]:

- a *depth*-2 TGC requires at least $m/\{n\log(m/n)\}$ TGs;
- a *depth*-3 TGC requires at least $2\,(m/\log m)^{1/2}$ TGs in each of the two hidden layer (if $m \gg n^2$);
- an arbitrarily interconnected TGC without feedback needs $(2m/\log m)^{1/2}$ TGs (if $m \gg n^2$).

One study[51] has tried to unify these two lines of research (*i.e.*, to find proofs for the universal approximation problem, while also bounding the *size*) by first presenting analytical solutions for the general NN problem in one dimension (having infinite *size*!), and then giving practical solutions for the one-dimensional cases (*i.e.*, including an upper bound on the *size*). Extensions to the n-dimensional case using three- and four-layers solutions were derived under piecewise constant approximations (having constant or variable width partitions), and under piecewise linear approximations (using ramps instead of sigmoids).

2.3 Boolean Functions ($f: \mathbb{B}^n \to \mathbb{B}$)

The particular case of BFs has been intensively studied.[17,52] Some results have been obtained for particular BFs,[43,44] but a *size*-optimal result for BFs that have

exactly m groups of ones in their truth table (equivalently, which are defined on the m groups) was detailed by Red'kin.[53]

Theorem[53] *The complexity realisation (i.e., number of threshold elements) of $I\!F_{n,m}$ is at most $2\,(2m)^{1/2} + 3$.*

All the previous mentioned results are valid for unlimited *fan-in* TGs. Departing from these lines, Horne and Hush[54] detail a solution for limited *fan-in* TGCs.

Theorem[54] *Arbitrary Boolean functions of the form $f: \{0, 1\}^n \to \{0, 1\}^m$ can be implemented in a neural network of perceptrons restricted to fan-in $\Delta = 2$ with a node complexity of $\Theta\,\{m\,2^n/(n + \log m)\}$ and requiring $O\,(n)$ layers.*

3 Analog Implementation of Boolean Functions

It is known that implementing any BFs using classical Boolean gates (i.e., AND and OR gates) requires exponential *size* circuits. As has been seen from all previous results, the known bounds for *size* are also exponential if TGCs are used for solving arbitrary BFs[55]. It is true that these bounds reveal exponential gaps (thus encouraging research efforts to reduce them), and also suggest that TGCs with more layers (*depth* $\neq$ small constant[56,57]) might have a smaller *size*.

A completely different approach is to use Kolmogorov's superpositions theorem, which shows that there are NNs having only $2n + 1$ neurons which can approximate any function. Such a solution would clearly be *size*-optimal. We start from[37-39], where a constructive solution for the general case was detailed.

Theorem[37] *Define the function $\psi: \mathcal{S} \to \mathcal{D}$ such that for each integer $k \in N$.*

$$\psi\left(\sum_{r=1}^{k} i_r \gamma^{-r}\right) = \sum_{r=1}^{k} \tilde{\imath}_r 2^{-m_r} \gamma^{-\frac{n^{r-m_r}-1}{n-1}} \tag{6}$$

where

$$\tilde{\imath}_r = i_r - (\gamma - 2)\langle i_r \rangle \tag{7}$$

and

$$m_r = \langle i_r \rangle \left(1 + \sum_{s=1}^{r-1} [i_s] \times \ldots \times [i_{r-1}]\right) \tag{8}$$

for $r = 1, 2, \ldots, k$.

Here $\gamma \geq 2n + 2$ is a base, $\mathcal{S} = [0, 1]$, $\mathcal{D}$ is the set of terminating rational numbers $d_k = \sum_{r=1}^{k} i_r \gamma^{-r}$ defined on $k \in N$ digits ($0 \leq i_r \leq \gamma - 1$). Also, $\langle i_1 \rangle = 0$ and $[i_1] = 0$, while for $r \geq 2$: $\langle i_r \rangle = 0$ when $i_r = 0, 1, \ldots, \gamma - 2$, and $\langle i_r \rangle = 1$ when $i_r = \gamma - 1$, $[i_r] = 0$ when $i_r = 0, 1, \ldots, \gamma - 3$, and $[i_r] = 1$ when $i_r = \gamma - 2, \gamma - 1$.

For BFs, one digit is enough ($k = 1$), which gives $\psi\,(0.i_1) = 0.i_1$ (or $\psi\,(x) = x$), and shows that the nonlinearity is the identity function.

Such a solution builds analog neurons having *fan-in* $\Delta \leq 2n + 1$, for which the known *weight* bounds[52,58–60] (holding for any *fan-in* $\Delta \geq 4$) are:

$$2^{(\Delta-1)/2} < weight < (\Delta+1)^{(\Delta+1)/2}/2^{\Delta}. \qquad (9)$$

Thus, one would expect to have a *precision* of between Δ and $\Delta \log \Delta$ bits per *weight*. Unfortunately, the solution for Kolmogorov's superposition[37–39] requires (in general) a double exponential precision for ψ (Eq. 6), and for the *weights*:

$$\alpha_p = \sum_{r=1}^{\infty} \gamma^{-(p-1)\frac{n^r-1}{n-1}}. \qquad (10)$$

For BFs this precision is reduced to $(2n+2)^{-n}$, *i.e.* $2n \log n$ bits per *weight*. Analog implementations are limited to just several bits of precision,[61] this being one of the reasons for investigations on required precision[62–65] and on algorithms relying on limited integer *weights*.[16,66–70]

An 'optimal' solution for implementing BFs should decompose the given function in simpler BFs which can be efficiently implemented based on Kolmogorov's superpositions (*i.e.*, we have to reduce n to small values). The partial results from this first layer of analog building blocks can be combined using again Kolmogorov's superpositions. The final implementation is analog, but requires more layers (for accommodating the limited precision of present day technologies).

4 Conclusions

Arbitrary BFs can be implemented using:
- classical Boolean gates, but require exponential *size*;
- TGs, but (again) in exponential *size* (still, there are exponential gaps between classical Boolean solutions and TG ones);
- analog building blocks in linear *size* (having linear *fan-in* and polynomial precision *weights* and *thresholds*); the nonlinear activation function is the identity function.

The main conclusion is that *size*-optimal hardware implementations of BFs can be obtained only in analog circuitry. The high precision required by the solution based on Kolmogorov's superpositions can be tackled by decomposing a BFs into simpler BFs.

References

1. M.A. Arbib, *The Handbook of Brain Theory and Neural Networks* (MIT Press, Cambridge, MA, 1995).
2. E. Fiesler and R. Beale, *Handbook of Neural Computation*, eds. (Oxford Univ. Press and the Inst. of Physics, NY, 1996).

3. E. Blum and K. Li, *Neural Networks* **4**, 511 (1991).
4. Y. Ito, *Neural Networks* **4**, 817 (1991).
5. Y. Ito, *Neural Computation* **6**, 1233 (1994).
6. K.-I. Funahashi and Y. Nakamura, *Neural Networks* **6**, 801 (1993).
7. R. Hecht-Nielsen, *Proc. ICNN'87* (IEEE Press, CA), 11 (1987).
8. R.P. Lippmann, *IEEE ASSP Mag.* **4**, 4 (1987).
9. Y. LeCun, *MSc Thesis*, Université Pierre et Marie Curie, Paris, France (1987).
10. D.A. Sprecher, *Trans. American Math. Soc.* **115**, 340 (1965).
11. D.A. Sprecher, *Proc. American Math. Soc.* **17**, 98 (1966).
12. A.N. Kolmogorov, *Dokl. Akad. Nauk SSSR* **114**, 953 (1957); English transl. *Transl. Amer. Math. Soc.* **2**, 55 (1963).
13. G. Cybenko, *Tech. Rep.*, Dept. of CS, Tufts Univ., Medford, MA (1988).
14. G. Cybenko, *Math. of Control, Signals and Systems* **2**, 303 (1989).
15. B. Irie and S. Miyake, *Proc. ICNN'88* (IEEE Press, CA), vol. **1**, 641 (1988).
16. F. Scarselli and A.C. Tsoi, *Neural Networks* **11**, 15 (1998).
17. V. Beiu, *VLSI Complexity of Discrete Neural Networks* (Gordon and Breach, Newark, NJ, 1998).
18. K.-I. Funahashi, *Neural Networks* **2**, 183 (1989).
19. K. Hornik *et al.*, *Neural Network* **2**, 359 (1989).
20. K. Hornik *et al.*, *Neural Network* **3**, 551 (1990).
21. J. Park and I.W. Sandberg, *Neural Computation* **3**, 246 (1991).
22. J. Park and I.W. Sandberg, *Neural Computation* **5**, 305 (1993).
23. K. Hornik, *Neural Network* **4**, 251 (1991).
24. S. Geva and J. Sitte, *IEEE Trans. Neural Networks* **3**, 621 (1992).
25. V. Kůrková, *Neural Networks* **5**, 501 (1992).
26. H. Mhaskar and C. Micchelli, *Advs. in Appl. Maths.* **13**, 350 (1992).
27. H. Mhaskar and C. Micchelli, *IBM J. Res. and Develop.* **38**, 277 (1994).
28. P. Koiran, *Neural Networks* **6**, 649 (1993).
29. M. Leshno *et al.*, *Neural Networks* **6**, 861 (1993).
30. K. Hornik, *Neural Network* **6**, 1069 (1993).
31. D.A. Sprecher, *Neural Networks* **6**, 1089 (1993).
32. A.R. Barron, *IEEE Trans. Info. Theory* **39**, 930 (1993).
33. L.K. Jones, *Ann. Statist.* **20**, 608 (1992).
34. H. Katsuura and D.A. Sprecher, *Neural Networks* **7**, 455 (1994).
35. M. Nees, *J. Comp. and Appl. Math.* **54**, 239 (1994).
36. M. Nees, *Advs. in Comp. Maths.* **5**, 137 (1996).
37. D.A. Sprecher, *Neural Networks* **9**, 765 (1996).
38. D.A. Sprecher, *Neural World* **6**, 711 (1996).
39. D.A. Sprecher, *Neural Networks* **10**, 447 (1997).
40. R.C. Minnik, *IRE Trans. Electr. Comp.* **10**, 6 (1961).
41. E.I. Neciporuk, *Soviet Mathematics—Doklady* **5**(1), 163 (1964); English transl. *Automation Express* **7**, 35 and 27 (1964).
42. O.B. Lupanov, *Problemy Kibernetiki* **20**, 109 (1973).
43. K.-Y. Siu *et al.*, *IEEE Trans. Comp.* **40**, 1402 (1991).

44. V.P. Roychowdhury *et al.*, *IEEE Trans. Info. Th.* **40**, 467 (1994).

45. E.B. Baum, *J. Complexity* **4**, 193 (1988).

46. S.-C. Huang and Y.-F. Huang, *IEEE Trans. Neural Networks* **2**, 47 (1991).

47. M. Arai, *Neural Networks* **6**, 855 (1993).

48. V. Beiu and T. De Pauw, *Tech. Rep. LA-UR-97-60* (Los Alamos Natl. Lab.), in *Biological and Artificial Computation*, eds. J. Mira *et al.* (LNCS 1240, Springer, Berlin), 743 (1997).

49. M.H. Hassoun, *Fundamentals of Artificial Neural Networks* (MIT Press, Cambridge, MA, 1995).

50. H. Paugam-Moisy, *PhD Dissertation*, LIP-IMAG, École Normale Supérieure de Lyon, France (1992).

51. A. Bulsari, *Neural Networks* **6**, 991 (1993).

52. I. Parberry, *Circuit Complexity and Neural Networks* (MIT Press, Cambridge, MA, 1994).

53. N.P. Red'kin, *Kibernetika* **6**, 6 (1970); English transl. *Cybernetics* **6**, 540 (1973).

54. B.G. Horne and D.R. Hush, *Neural Networks* **7**, 1413 (1994).

55. V. Beiu, Chapter E1.4^2 (1996).

56. V. Beiu, *Tech. Rep. LA-UR-96-3591* (Los Alamos Natl. Lab.), *Neurocomputing* **19**, 77 (1998).

57. V. Beiu and H.E. Makaruk, *Tech. Rep. LA-UR-97-4413* (Los Alamos Natl. Lab.), to appear in *Neural Proc. Lett.* **7/8** (1998).

58. J. Myhill and W.H. Kautz, *IRE Trans. Electr. Comp.* **10**, 288 (1961).

59. P. Raghavan, *Tech. Rep. RC-13859*, IBM Res. (1988); also in *Proc. COLT'88* (ACM Press, NY), 19 (1988).

60. E.D. Sontag, *Tech. Rep. SYCON 96-01*, Dept. of Maths., Rutgers Univ. (1996); also in *Neural Computation* **9**, 337 (1997).

61. A.H. Kramer, *Proc. MicroNeuro'96* (IEEE Press, CA), 68 (1996).

62. J.S. Denker and B.S. Wittner in *Neural Information Processing Systems*, ed. D.Z. Anderson (American Inst. of Physics, NY), 219 (1988).

63. J.L. Holt and J.-N. Hwang, *IEEE Trans. Comp.* **42**, 281 (1993).

64. J. Wray and G.G.R. Green, *Neural Networks* **8**, 31 (1995).

65. M. Stevenson and S. Huq, *Neural Computation* **8**, 1603 (1996).

66. A.H. Khan and E.L. Hines, *Electronic Lett.* **30**, 1237 (1994).

67. S. Draghici and I.K. Sethi in *Biological and Artificial Computation*, eds. J. Mira *et al.* (LNCS 1240, Springer, Berlin), 753 (1997).

68. V. Beiu and S. Draghici, *Tech. Rep. LA-UR-97-294* (Los Alamos Natl. Lab.), in *Proc. SOCO'97*, ed. D.W. Pearson (ICSC Acad. Press, Canada), 111 (1997).

69. V. Beiu, S. Draghici and T. De Pauw, *Tech. Rep. LA-UR-97-884* (Los Alamos Natl. Lab.), to appear in *Neural Proc. Lett.* **7/8** (1998).

70. V. Beiu, *Tech. Rep. LA-UR-97-515* (Los Alamos Natl. Lab.), *Neural Network World* **8**, 1 (1998).