CONF-971217--

LA-UR- 97-1916

Title: DEEPER SPARSELY NETS ARE SIZE-OPTIMAL

RECEIVED
AUG 13 1997
OSTI

Author(s): VALERIU BEIU, NIS-1
HANNA MAKARUK, T-13

Submitted to: NEURAL INFORMATION PROCESSING SYSTEMS
DENVER, COLORADO
DECEMBER 1-6, 1997

Los Alamos
NATIONAL LABORATORY

DISCLAIMER

Deeper Sparsely Nets Are Size-Optimal *

Valeriu Beiu [1] **Hanna E. Makaruk** [2]
Space & Atmospheric Div. NIS-1, MS D466 Theoretical Div. T-13 MS B213
Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA

Abstract

The starting points of this paper are two *size*-optimal solutions: (*i*) one for implementing arbitrary Boolean functions (Horne, 1994); and (*ii*) another one for implementing certain sub-classes of Boolean functions (Red'kin, 1970). Because VLSI implementations do not cope well with highly interconnected nets—the *area* of a chip grows with the cube of the *fan-in* (Hammerstrom, 1988)—this paper will analyse the influence of limited *fan-in* on the *size* optimality for the two solutions mentioned. First, we will extend a result from Horne & Hush (1994) valid for *fan-in* $\Delta = 2$ to arbitrary *fan-in*. Second, we will prove that *size*-optimal solutions are obtained for small constant *fan-in* for both constructions, while relative minimum *size* solutions can be obtained for *fan-ins* strictly lower that linear. These results are in agreement with similar ones proving that for small constant *fan-ins* ($\Delta = 6 \ldots 9$) there exist VLSI-optimal (*i.e.*, minimising AT^2) solutions (Beiu, 1997a), while there are similar small constants relating to our capacity of processing information (Miller 1956).

1 INTRODUCTION

In this paper we shall consider feedforward neural networks (NNs) made of linear threshold gates (TGs), or perceptrons. A TG is computing a Boolean function (BF) $f\colon \{0, 1\}^n \rightarrow \{0, 1\}$, where an input vector is $\mathbf{Z}_k = (z_{k,0}, \ldots, z_{k,n-1})$ and $f(\mathbf{Z}_k) = sgn\,(\sum_{i=0}^{n-1} w_i z_{k,i} + \theta)$, with the synaptic *weights* $w_i \in \mathbb{R}$, $\theta \in \mathbb{R}$ known as the *threshold*, and *sgn* the sign function. The cost functions commonly associated are *depth* (*i.e.*, number of edges on the longest input to output path, or number of layers) and *size* (*i.e.*, number of neurons). However, the *area* of the connections counts, and the *area* of one neuron can be related to its associated *weights*, thus *"comparing the number of nodes is inadequate for comparing the complexity of NNs as the nodes themselves could implement quite complex functions"* (Williamson, 1990). That is why several authors (Abu-Mostafa, 1988; Hammerstrom, 1988; Phatak, 1994) have taken into account the total *number-of-connections*, others (Bruck, 1988) the total *number-of-bits* needed to represent the *weights* and the *thresholds*, or the sum of all the *weights* and the *thresholds* (Beiu, 1994). This measure (also applied for defining the minimum-integer TG realisation of a BF) has been recently used—under the name of *"total*

* Or ***"Small Fan-In is Beautiful."***

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

[2] On leave of absence from the Polish Academy of Sciences, Institute of Fundamental Technological Research, Świetokrzyska 21, 00-049 Warsaw, Poland.

weight magnitude"—in the context of computational learning theory for improving on several standard VC-theory bounds (Bartlett, 1996). A quite similar definition of 'complexity' $\sum w_i^2$ has also been advocated (Zhang, 1993). Such approximations can easily be related to assumptions on how the *area* of a chip scales with the *weights* and the *thresholds* (Beiu, 1996b, 1997a):

- for digital implementation the *area* scales with the cumulative size of the *weights* and *thresholds* (as the bits for representing those *weights* and *thresholds* have to be stored);
- for analog implementations (*e.g.*, using resistors or capacitors) the same type of scaling is valid (although it is possible to come up with implementations having binary encoding of the parameters—for which the *area* would scale with the cumulative log-scale size of the parameters);
- some types of implementations (*e.g.*, transconductance ones) even offer a constant size per element, thus in principle scaling only with the number of parameters (*i.e.*, with the total *number-of-connections*).

It is worth emphasising that it is anyhow desirable to limit the range of parameter values (Wray, 1995) for VLSI implementations because: (*i*) the maximum value of the *fan-in* (Walker, 1989); and (*ii*) the maximal ratio between the largest and the smallest *weight* cannot grow over a certain (technological) limit. The paper will discuss the influence of limiting the *fan-in* on the *size* optimality of two different *size*-optimal solutions, and is structured as follows: in Section 2 we present pervious results, while in Section 3 we shall prove our main claims. Conclusions and open problems for research are ending the paper.

2 PREVIOUS RESULTS

One starting point is a classic construction for synthesising one BF with *fan-in* 2 AND-OR gates. It was extended to the multioutput case and modified to apply to NNs.

Proposition 1 (Theorem 3 from Horne 1994) *Arbitrary Boolean functions of the form* $f: \{0,1\}^n \rightarrow \{0,1\}^m$ *can be implemented in a NN of perceptrons restricted to fan-in 2 with a node complexity of* $\Theta\{m2^n/(n+\log m)\}$ *and requiring* $O(n)$ *layers.*

Sketch of proof The idea is to decompose each output BF into two subfunctions using Shannon's Decomposition $f(x_1 x_2 \ldots x_{n-1} x_n) = \bar{x}_1 f_0(x_2 \ldots x_{n-1} x_n) + x_1 f_1(x_2 \ldots x_{n-1} x_n)$. By doing this recursively for each subfunction, the output BFs will—in the end—be implemented by binary trees. Horne & Hush (1994) use a trick for eliminating most of the lower level nodes by replacing them with a subnetwork that computes *all the possible BFs* needed by the higher level nodes. Each subcircuit eliminates one variable and has three nodes (one OR and two ANDs), thus the upper tree has:

$$size_{upper} = 3 \cdot \sum_{i=0}^{n-q-1} 2^i = 3(2^{n-q}-1) \tag{1}$$

nodes and $depth_{upper} = 2(n-q)$. The subfunctions now depend on only q variables, and a lower subnetwork that computes all the possible BFs of q variables is built. It has:

$$size_{lower} = 3 \cdot \sum_{i=1}^{q} 2^{2^i} < 4 \cdot 2^{2^q} \tag{2}$$

nodes and $depth_{lower} = 2(n-q)$ (see Figure 2 in (Horne, 1994)). That q which minimises $size_{BFs} = size_{upper} + size_{lower}$ is determined by solving $d(size_{BFs})/dq = 0$, and gives:

$$q \approx \log\{n + \log\mu - 2\log(n+\log\mu)\}. \tag{3}$$

By substituting (3) in (1) and (2), the minimum $size_{BFs}$ can be determined. ❑

Proposition 2 (Theorem 1 from Red'kin 1970) *The complexity realisation (i.e., number of threshold elements) of* $\mathbb{F}_{n,m}$ *(the class of Boolean functions* $f(x_1 x_2 \ldots x_{n-1} x_n)$ *that have exactly m groups of ones) is at most* $2(2m)^{1/2}+3$.

The construction has: a first layer of $\lceil (2m)^{1/2} \rceil$ TGs (COMPARISONs) with *fan-in* $= n$ and *weights* $\leq 2^{n-1}$; a second layer of $2\lceil (m/2)^{1/2} \rceil$ TGs of *fan-in* $= n + \lceil (2m)^{1/2} \rceil$ and *weights* $\leq 2^n$; one more TG of *fan-in* $= 2\lceil (m/2)^{1/2} \rceil$ and *weights* $\in \{-1, +1\}$ in the third layer.

3 LIMITED FAN-IN AND OPTIMAL SOLUTIONS

Proposition 3 (this paper) *Arbitrary Boolean functions* $f: \{0, 1\}^n \rightarrow \{0, 1\}^m$ *can be implemented in a NN of perceptrons restricted to fan-in* Δ *in* $O(n/\log\Delta)$ *layers.*

Proof We use the same approach as Horne & Hush (1994) for the case when the *fan-in* is limited to Δ. Each output BF can be decomposed in $2^{\Delta-1}$ subfunctions (*i.e.*, $2^{\Delta-1}$ AND gates). The OR gate would have $2^{\Delta-1}$ inputs, thus we have to decompose it in a Δ-ary tree of *fan-in* $= \Delta$ OR gates. This decomposition step eliminates $\Delta - 1$ variables and generates a $depth = 1 + \lceil(\Delta-1)/\log\Delta\rceil$, and $size = 2^{\Delta-1} + \lceil(2^{\Delta-1}-1)/(\Delta-1)\rceil$ Δ-ary tree. Repeating this procedure recursively k times, we have:

$$depth_{upper} = k \cdot \{1 + \lceil(\Delta-1)/\log\Delta\rceil\} \tag{4}$$

$$\begin{aligned} size_{upper} &= \{2^{\Delta-1} + \lceil(2^{\Delta-1}-1)/(\Delta-1)\rceil\} \cdot \textstyle\sum_{i=0}^{k-1} 2^{i(\Delta-1)} \\ &= size \cdot (2^{k(\Delta-1)}-1)/(2^{\Delta-1}-1) \\ &\cong 2^{k(\Delta-1)}(1+1/\Delta) \\ &\approx 2^{k\Delta-k} \end{aligned} \tag{5}$$

where the subfunctions depend only on $q = n - k\Delta$ variables. We now generate all the possible subfunctions of q variables with a subnetwork of:

$$depth_{lower} = \lfloor(n-k\Delta)/\Delta\rfloor \cdot \{1 + \lceil(\Delta-1)/\log\Delta\rceil\} \tag{6}$$

$$\begin{aligned} size_{lower} &= \{2^{\Delta-1} + \lceil(2^{\Delta-1}-1)/(\Delta-1)\rceil\} \cdot \textstyle\sum_{i=1}^{\lfloor n/\Delta\rfloor-k} 2^{2^{n-k\Delta-i\Delta}} \\ &= size \cdot \left\{ 2^{2^0} + 2^{2^\Delta} + \ldots + 2^{2^{n-(k+1)\Delta}} \right\} \\ &< (size+1) \cdot 2^{2^{n-(k+1)\Delta}} && (7) \\ &\approx 2^\Delta \cdot 2^{2^{n-k\Delta-\Delta}}. && (8) \end{aligned}$$

The inequality (7) can be proved by induction. Clearly, $size \cdot 2^{2^0} < (size+1) \cdot 2^{2^0}$. Let us consider the statement true for α; we prove it for $\alpha + 1$:

$$size \cdot \left\{ 2^{2^0} + 2^{2^\Delta} + \ldots + 2^{2^{\alpha\Delta}} \right\} + size \cdot 2^{2^{(\alpha+1)\Delta}} < size \cdot 2^{2^{(\alpha+1)\Delta}} + 2^{2^{(\alpha+1)\Delta}}$$

$$size \cdot \left\{ 2^{2^0} + 2^{2^\Delta} + \ldots + 2^{2^{\alpha\Delta}} \right\} < (size+1) \cdot 2^{2^{\alpha\Delta}}$$

(due to hypothesis), thus:

$$(size+1) \cdot 2^{2^{\alpha\Delta}} < 2^{2^{(\alpha+1)\Delta}}$$

and computing the logarithm of the left side:

$$\begin{aligned} 2^{\alpha\Delta} + \log(size+1) &= 2^{\alpha\Delta} + \log\{2^{\Delta-1} + \lceil(2^{\Delta-1}-1)/(\Delta-1)\rceil\} \\ &< 2^{\alpha\Delta} + \log\{2^{\Delta-1} + 2^{\Delta-1}/\Delta + 1\} \\ &< 2^{\alpha\Delta} + \Delta \\ &< 2^{(\alpha+1)\Delta}. \end{aligned}$$

From (4) and (6) we can estimate $depth_{BFs}$, and from (5) and (8) $size_{BFs}$ as:

$$\begin{aligned} depth_{BFs} &= \{k + \lfloor(n-k\Delta)/\Delta\rfloor\} \cdot \{1 + \lceil(\Delta-1)/\log\Delta\rceil\} \\ &= (n/\Delta) \cdot (\Delta/\log\Delta + 1) && (9) \\ &\approx n/\log\Delta = O(n/\log\Delta) \end{aligned}$$

$$\begin{aligned} size_{BFs} &= m \cdot size \cdot (2^{k(\Delta-1)}-1)/(\Delta-1) + (size+1) \cdot 2^{2^{n-(k+1)\Delta}} \\ &\approx m \cdot 2^{k\Delta-k} + 2^\Delta \cdot 2^{2^{n-k\Delta-\Delta}} && (10) \end{aligned}$$

concluding the proof. ❑

Proposition 4 (this paper) *All the critical points of* $size_{BFs}(m, n, k, \Delta)$ *are relative minimum and are situated in the (close) vicinity of the parabola* $k\Delta \approx n - \log(n + \log m)$.

Proof To determine the critical points we equate the partial derivatives to zero. Starting from the approximation of $size_{BFs}$ we compute $\partial size_{BFs} / \partial k = 0$:

$$m \cdot 2^{k\Delta - k} (\ln 2)(\Delta - 1) + 2^{\Delta} \cdot 2^{2^{n - k\Delta - \Delta}} (\ln 2) \cdot 2^{n - k\Delta - \Delta} (\ln 2) \cdot (-\Delta) = 0$$

$$\{m(\Delta - 1)/\Delta/(\ln 2)\} \cdot 2^{2k\Delta - k - n} = 2^{2^{n - k\Delta - \Delta}}$$

and using the notations $k\Delta = \gamma$, $\beta = m(\Delta - 1)/(\Delta \ln 2)$, and taking logarithms of both sides:

$$\log\beta + 2\gamma - k - n = 2^{n - \gamma - \Delta} \tag{11}$$

which has an approximate solution $\gamma \approx n - \log(n + \log m)$. The same result can be obtained by computing with finite differences (instead of approximating the partial derivative):

$$size_{BFs}(m, n, k+1, \Delta) - size_{BFs}(m, n, k, \Delta) = 0$$

$$size \cdot \left\{ m \cdot 2^{k\Delta - k} - 2^{2^{n - k\Delta - \Delta}} \right\} = 0$$

$$m \cdot 2^{k\Delta - k} = 2^{2^{n - k\Delta - \Delta}}$$

and after taking twice the logarithm of both sides and using the same notations we have:

$$\log\{\log m + \gamma(1 - 1/\Delta)\} = n - \gamma - \Delta$$

$$\begin{aligned} \gamma &= n - \{\Delta + \log(1 - 1/\Delta)\} - \log\{\gamma + \Delta/(\Delta - 1) \cdot \log m\} \\ &\approx n - \Delta - \log(\gamma + \log m), \end{aligned} \tag{12}$$

which has as approximate solution $\gamma = n - \log(n + \log m)$.

Starting again from (10) we compute $\partial size_{BFs} / \partial\Delta = 0$:

$$m\, 2^{k\Delta - k} (\ln 2)\, k + 2^{\Delta} (\ln 2)\, 2^{2^{n - k\Delta - \Delta}} + 2^{\Delta}\, 2^{2^{n - k\Delta - \Delta}} (\ln 2)\, 2^{n - k\Delta - \Delta} (\ln 2)(-k) = 0$$

$$\begin{aligned} mk \cdot 2^{\gamma - k} &= k(\ln 2) \cdot 2^{n - \gamma} \cdot 2^{2^{n-\gamma-\Delta}} - 2^{\Delta} \cdot 2^{2^{n-\gamma-\Delta}} \\ mk \cdot 2^{\gamma - k} \cdot 2^{\gamma - n} &= k(\ln 2) \cdot 2^{2^{n-\gamma-\Delta}} - 2^{\Delta} \cdot 2^{\gamma - n} \cdot 2^{2^{n-\gamma-\Delta}} \\ mk \cdot 2^{2\gamma - k - n} &= \{k(\ln 2) - 2^{\gamma + \Delta - n}\} \cdot 2^{2^{n-\gamma-\Delta}} \\ (m/\ln 2) \cdot 2^{2\gamma - k - n} &= \left\{1 - 2^{\gamma + \Delta - n}/(k \ln 2)\right\} \cdot 2^{2^{n-\gamma-\Delta}} \end{aligned}$$

which—by neglecting $2^{\gamma + \Delta} / \{k(\ln 2) \cdot 2^{n}\}$—gives:

$$\log\beta + 2\gamma - k - n = 2^{n - \gamma - \Delta}$$

i.e., the same equation as (11). These show that the critical points are situated in the (close) vicinity of the parabola $k\Delta \approx n - \log(n + \log m)$. The fact that they are relative minimum has also been proven (Beiu 1997b). ❑

The exact *size* has been computed for many different values of n, m, Δ and k. One example of those extensive simulations is plotted in Figure 1. From Figure 1(a) it may seem that k and Δ have almost the same influence on $size_{BFs}$. The discrete parabola-like curves (the one closer to the axes is approximating $k\Delta \approx n - \log(n + \log m)$) can be seen in Figure 1(b).

Proposition 5 (this paper) *The absolute minimum* $size_{BFs}$ *is obtained for fan-in* $\Delta = 2$.

Sketch of proof We will analyse only the critical points by using the approximation $k\Delta \approx n - \log n$. Intuitively the claim can be understood if we replace this value in (10):

$$\begin{aligned} size^{*}_{BFs} &\approx m \cdot 2^{n - \log n - k} + 2^{\Delta} \cdot 2^{2^{n - n + \log n - \Delta}} \\ &< m \cdot 2^{n - \log n} + 2^{\Delta} \cdot 2^{2^{\log n}} \\ &= m \cdot 2^{n}/n + 2^{\Delta} \cdot 2^{n}, \end{aligned}$$

which clearly is minimised for $\Delta = 2$. ❑

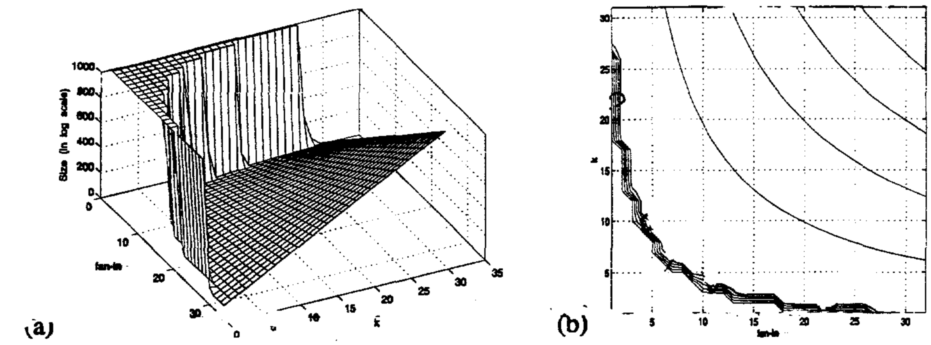

Figure 1: (a) Exact *size* as a function of the *fan-in* Δ and k, for $n = 64$ and $m = 1$; (b) contour plot.

The detailed proof relies on computing $size_{BFs}(n, m, k, \Delta)$ for $k \approx (n - \log n) / \Delta$, and then showing that $size^*_{BFs}(n, m, \Delta + 1) - size^*_{BFs}(n, m, \Delta) > 0$, thus the function being monotonically increasing and the minimum is obtained for the smallest *fan-in* $\Delta = 2$. Because the proof has been obtained using successive approximations, several simulation results are presented in Table 1. It can be seen that while for relatively small n the *size*-optimal solutions are obtained even for $\Delta = 16$, starting from $n \geq 64$ all the *size*-optimal solutions are obtained for $\Delta = 2$. It is to be mentioned that the other relative minimum (on, or in the vicinity of the parabola $k\Delta \approx n - \log n$) are slightly larger than the absolute minimum. They might be of *practical interest* as leading to networks having fewer layers: $n / \log\Delta$ instead of n. Last, but not least, it is to be remarked that all these relative minimum are obtained for *fan-ins* strictly lower that linear (as $\Delta \leq n - \log n$).

A similar result can be obtained for $\mathbb{F}_{n,m}$, because the first layer is represented by COMPARISONs (*i.e.*, $\mathbb{F}_{n,1}$) which can be decomposed for satisfying the limited *fan-in* condition.

Proposition 6 (Lemma 1 & Corollary 1 from Beiu, 1994) *The* COMPARISON *of two n-bit numbers can be computed by a Δ-ary tree NN having integer weights and thresholds bounded by $2^{\Delta/2}$ for any $3 \leq \Delta \leq n$.*

The *size* complexity of the NN implementing one $\mathbb{F}_{n,m}$ function is (Beiu, 1994):

$$size_{\mathbb{F}} = 2nm \cdot \left\{ \frac{1}{\Delta/2} + \ldots + \frac{1}{(\Delta/2)^{depth_{\mathbb{F}}}} \right\}, \qquad (13)$$

where $depth_{\mathbb{F}} = \lceil \log n / (\log\Delta - 1) \rceil$, but a substantial enhancement is obtained if the *fan-in* is limited. The maximum number of *different BFs* which can be computed in each layer is:

$$(2n/\Delta)\, 2^{\Delta}, \qquad \frac{2n/\Delta}{\Delta/2}\, 2^{\Delta(\Delta/2)}, \ \ldots, \ \frac{2n/\Delta}{(\Delta/2)^{depth_{\mathbb{F}} - 1}} \cdot 2^{\Delta(\Delta/2)^{depth_{\mathbb{F}} - 1}}. \qquad (14)$$

For large m (needed for achieving a certain precision), and/or large n, the first terms of the sum (13) will be larger than the equivalent ones from (14). This is equivalent to the trick from (Horne, 1994), as the lower levels will compute *all the possible functions* realisable using only limited *fan-in* COMPARISONs. The optimum *size* becomes:

Table 1.

Minimum $size_{BFs}$ for different values of n and $m = 1$.

n	$8 = 2^3$	$16 = 2^4$	$32 = 2^5$	$64 = 2^6$	$128 = 2^7$	$256 = 2^8$	$512 = 2^9$	$1024 = 2^{10}$	$2048 = 2^{11}$
size	110	1470	349,530	1.611×10^{9}	6.917×10^{18}	5.104×10^{38}	2.171×10^{76}	1.005×10^{154}	1.685×10^{307}
Δ	**4**	**8**	**16**	**2**	**2**	**2**	**2**	**2**	**2**
$k\Delta$	4	8	16	58	122	248	504	1014	2038

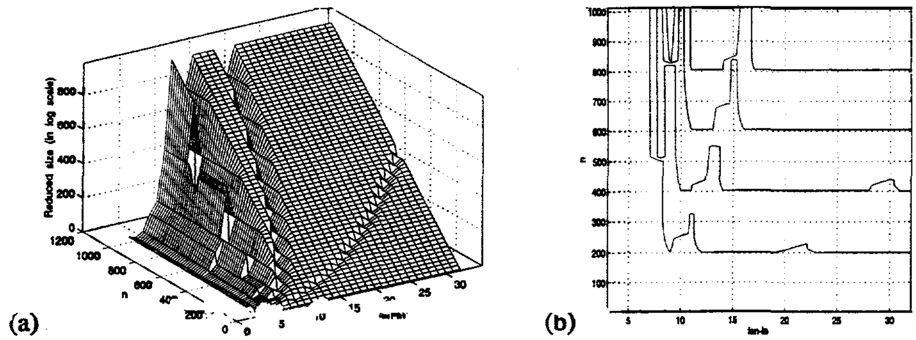

Figure 2: (a) *Size* of NNs for $\mathbb{F}_{n,m}$ when $m = 2^{0.99n}$ (almost completely specified); (b) contour plot.

$$size^*_{\mathbb{F}} = 2n \cdot \left\{ \sum_{i=1}^{k} \frac{2^{\Delta(\Delta/2)^{i-1}}}{\Delta(\Delta/2)^{i-1}} + \sum_{i=k+1}^{depth_{\mathbb{F}}} \frac{m}{(\Delta/2)^i} \right\}.$$

Following similar steps to the ones used in *Proposition 5*, it is possible to show that the minimum *size* is obtained for $\Delta = 3$. To get a better understanding we have done simulations by considering that $m = 2^{\varepsilon n}$. Some results can be seen in Figure 2 (for $\varepsilon = 0.99$).

We mention here that similar results ($\Delta = 6\ldots9$), based on closer estimates of *area* and *delay* have been proven for VLSI-efficient implementations of $\mathbb{F}_{n,m}$ functions (Beiu 1996b, 1997a). Different complexity estimates for COMPARISON can be seen in Table 2. All of these support the claim that small constant *fan-in* NNs can be *size*- and VLSI-optimal.

4 CONCLUSIONS AND OPEN PROBLEMS

In this paper we have extended a result from Horne & Hush (1994) valid for *fan-in* $\Delta = 2$ to arbitrary *fan-ins*, and have shown that the minimum *size* is obtained for small

Table 2 (from Beiu 1996b).

Different estimates of AT^2 for SRK (Siu, 1991), B_4 and B_log (Beiu, 1994, 1996b), ROS (Roychowdhury, 1994) and VCB (Vassiliadis, 1996).

Delay / *Area*	*Depth*	*Fan-in*	*Length*
Size	$AT^2_{VCB} = O(\sqrt{n})$ $AT^2_{ROS} = O(n/\log n)$ $AT^2_{SRK} = O(n)$ $AT^2_{B\_log} = O[n \log n / \log^2(\log n)]$ $AT^2_{B\_4} = O(n \log^2 n)$	$AT^2_{B\_4} = O(n \log^2 n)$ $AT^2_{B\_log} = O[n \log^3 n / \log^2(\log n)]$ $AT^2_{VCB} = O(n\sqrt{n})$ $AT^2_{ROS} = O(n^3/\log^3 n)$ $AT^2_{SRK} = O(n^3)$	$AT^2_{VCB} = O(n^2\sqrt{n})$ $AT^2_{ROS} \cong 3 \cdot n^3/\log n$ $AT^2_{B\_log} \cong 4 \cdot n^3/\log n$ $AT^2_{B\_4} \cong 4 \cdot n^3$ $AT^2_{SRK} \cong 27n^3/4$
$\sum_{NN}$ *fan-ins*	$AT^2_{VCB} = O(n)$ $AT^2_{B\_log} = O[n \log^2 n / \log^2(\log n)]$ $AT^2_{B\_4} = O(n \log^2 n)$ $AT^2_{ROS} = O(n^2/\log^2 n)$ $AT^2_{SRK} = O(n^2)$	$AT^2_{B\_4} = O(n \log^2 n)$ $AT^2_{B\_log} = O[n \log^4 n / \log^2(\log n)]$ $AT^2_{VCB} = O(n^2)$ $AT^2_{ROS} = O(n^4/\log^4 n)$ $AT^2_{SRK} = O(n^4)$	$AT^2_{B\_log} \cong 4n^3$ $AT^2_{VCB} \cong 4n^3$ $AT^2_{B\_4} \cong 5n^3$ $AT^2_{ROS} = O(n^4/\log^2 n)$ $AT^2_{SRK} = O(n^4)$
$\sum_{NN}(\sum_i \lvert w_i \rvert + \lvert \theta \rvert)$	$AT^2_{B\_4} = O(n \log^2 n)$ $AT^2_{B\_log} = O[n\sqrt{n} \log n / \log^2(\log n)]$ $AT^2_{ROS} = O(n^2/\log n)$ $AT^2_{SRK} = O(n^2)$	$= O(n \log^2 n)$ $AT^2_{B\_log} = O\{n\sqrt{n} \log^3 n / \log^2(\log n)\}$ $AT^2_{ROS} = O(n^4/\log^3 n)$ $AT^2_{SRK} = O(n^4)$	$AT^2_{B\_4} = O(n^3)$ $AT^2_{B\_log} = O(n^3\sqrt{n}/\log n)$ $AT^2_{ROS} = O(n^4/\log n)$ $AT^2_{SRK} = O(n^4)$

(constant) *fan-ins*. We have also shown that, using their construction, it is possible to obtain 'good' (*i.e.*, relative minimum) solutions for *fan-ins* strictly lower than linear. The same results have been obtained for the *size*-optimal solution of Red'kin (1970). The main conclusions are that: (*i*) there are interesting *fan-in* dependent *depth-size* (and *area-delay*) tradeoffs; and (*ii*) there are optimal solutions having small constant *fan-in* values. Future work is concentrating on linking these results with the entropy of the data-set, and with principles like the "Occam's razor" (Zhang, 1993) and the "minimum description length", as well as trying to find closer estimates for mixed analogue/digital implementations.

References

Abu-Mostafa, Y.S. (1988) Connectivity Versus Entropy. In D.Z. Anderson (ed.), *NIPS*, 1-8. New York, NY: AIP.

Bartlett, P.L. (1996) The Sample Complexity of Pattern Classification with Neural Networks: The Size of the Weights Is More Important than the Size of the Network. *Tech. Rep.*, Dept. Sys. Eng., Australian Natl. Univ., Canberra [short version in (1997) *NIPS 9*, Cambridge, MA: MIT Press].

Beiu, V., Peperstraete, J.A., Vandewalle, J. & Lauwereins, R. (1994) Area-Time Performances of Some Neural Computations. In P. Borne *et al.* (eds.): *SPRANN'94*, 664-668, Lille: GERF EC.

Beiu, V. & Taylor, J.G. (1996a) On the Circuit Complexity of Sigmoid Feedforward Neural Networks. *Neural Networks* 9(7):1155-1171.

Beiu, V. (1996b) On the Circuit and VLSI Complexity of Threshold Gate COMPARISON. *Tech. Rep. LA-UR-96-3591*, Los Alamos Natl. Lab., USA [(1997), *Neurocomputing*, to appear].

Beiu V. (1997a) Constant Fan-In Digital Neural Networks Are VLSI-Optimal. *Tech. Rep. LA-UR-97-61*, Los Alamos Natl. Lab., USA [in S.W. Ellacott *et al.* (eds.), *Mathematics of Neural Nets: Models, Algorithms & Applications*, Boston, MA: Kluwer, to appear].

Beiu, V. (1997b) When Constants Are Important. *Tech. Rep. LA-UR-97-226*, Los Alamos Natl. Lab., USA (to appear in *Proc. Intl. Conf. Control Sys. & Comp. Sci. CSCS-11*, Bucharest, Romania).

Beiu, V. (1997c) *VLSI Complexity of Discrete Neural Networks*. Newark, NJ: Gordon & Breach.

Bruck, J. & Goodmann, J.W. (1988) On the Power of Neural Networks for Solving Hard Problems. In D.Z. Anderson (ed.), *NIPS*, 137-143. New York, NY: AIP [(1990), *J. Complexity* 6:129-135].

Hammerstrom, D. (1988) The Connectivity Analysis of Simple Association –or– How Many Connections Do You Need. In D.Z. Anderson (ed.), *NIPS*, 338-347. New York, NY: AIP.

Horne, B.G. & Hush, D.R. (1994) On the Node Complexity of Neural Networks. *Neural Networks* 7(9):1413-1426.

Miller G.A. (1956) The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information. *Psych. Rev.* 63:71-97.

Phatak, D.S. & Koren, I. (1994) Connectivity and Performances Tradeoffs in the Cascade Correlation Learning Architecture. *IEEE Trans. Neural Networks* 5(6):930-935.

Red'kin, N.P. (1970) Synthesis of Threshold Circuits for Certain Classes of Boolean Functions. *Kibernetika* 5:6-9 [English translation (1973), *Cybernetics* 6(5):540-544].

Roychowdhury, V.P., Orlitsky, A. & Siu, K.-S. (1994) Lower Bounds on Threshold and Related Circuits Via Communication Complexity. *IEEE Trans. Information Theory* 40(2):467-474.

Siu, K.-Y., Roychowdhury, V.P. & Kailath, T. (1991) Depth-Size Tradeoffs for Neural Computations. *IEEE Trans. Comp.* 40(12):1402-1412.

Vassiliadis, S., Cotofana, S. & Berteles, K. (1996) 2-1 Addition and Related Arithmetic Operations with Threshold Logic. *IEEE Trans. Comp.* 45(9):1062-1068.

Walker, M.R., Haghighi, S., Afghan, A. & Akers, L.A. (1989) Training a Limited-Interconnect, Synthetic Neural IC. In D.S. Touretzky (ed.), *NIPS 1*, 777-784. San Mateo, CA: Morgan Kaufmann.

Williamson, R.C. (1990) ε-Entropy and the Complexity of Feedforward Neural Networks. In R.P. Lippmann *et al.* (eds.), *NIPS 3*, 946-952. San Mateo, CA: Morgan Kaufmann.

Wray, J. & Green, G.G.R. (1995) Neural Networks, Approximation Theory, and Finite Precision Computation. *Neural Networks* 8(1):31-37.

Zhang, B.-T. & Mühlenbein, H. (1993) Genetic Programming of Minimal Neural Networks Using Occam's Razor. *Tech. Rep. GMD 0734*, Schloß Birlinghoven, St. Augustin, Germany [(1993) *Complex Systems* 7(3):199-220].