LA-UR- 97-4314

58

CONF-971235--

TITLE: ON LIMITED FAN-IN OPTIMAL NEURAL NETWORKS

AUTHOR(S): Valeriu Beiu, NIS-1
Sorin Draghici, Wayne State Univ
Hanna Makaruk, T-13

19980507 057

SUBMITTED TO: IVTH BRASILIAN SYMPOSIUM ON NEURAL NETWORKS, SBRN 97, BRASIL, 3-5 DECEMBER 1997

Los Alamos

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

CONF-971235--

On Limited Fan-in Optimal Neural Networks

Valeriu Beiu [1, 2], Sorin Draghici [3] and Hanna E. Makaruk [4, 5]

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

[3] Vision and Neural Networks Laboratory, Department of Computer Science
Wayne State University, 431 State Hall, Detroit, MI 48202, USA
E-mail: sod@cs.wayne.edu

[4] Theoretical Division T-13, MS B213, Los Alamos National Laboratory
Los Alamos, New Mexico 87545, USA
E-mail: hanna@t13.lanl.gov

Abstract

Because VLSI implementations do not cope well with highly interconnected nets—the area of a chip growing as the cube of the fan-in [25]—this paper analyses the influence of limited fan-in on the size and VLSI optimality of such nets.

Two different approaches will show that VLSI- and size-optimal discrete neural networks can be obtained for small (i.e. lower than linear) fan-in values. They have applications to hardware implementations of neural networks. The first approach is based on implementing a certain sub-class of Boolean functions, $\mathbb{F}_{n,m}$ functions [34]. We will show that this class of functions can be implemented in VLSI-optimal (i.e., minimising AT^2) neural networks of small constant fan-ins. The second approach is based on implementing Boolean functions for which the classical Shannon's decomposition can be used. Such a solution has already been used to prove bounds on neural networks with fan-ins limited to 2 [26]. We will generalise the result presented there to arbitrary fan-in, and prove that the size is minimised by small fan-in values, while relative minimum size solutions can be obtained for fan-ins strictly lower than linear.

Finally, a size-optimal neural network having small constant fan-ins will be suggested for $\mathbb{F}_{n,m}$ functions.

Keywords—*neural networks, VLSI, fan-in, Boolean circuits, threshold circuits, $\mathbb{F}_{n,m}$ functions.*

[2] 'On leave of absence' from the "Politehnica" University of Bucharest, Computer Science Department, Spl. Independentei 313, RO–77206 Bucharest, România.

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

1. Introduction

In this paper we shall consider feedforward neural networks (NNs) made of linear threshold gates (TGs). A neuron (*i.e.*, linear TG) will compute a Boolean function (BF) $f:\{0,1\}^n \rightarrow \{0,1\}$, where one of the k input vectors is $\mathbf{Z}_k = (z_{k,0}, \ldots, z_{k,n-1}) \in \{0,1\}^n$ and $f(\mathbf{Z}_k) = sgn\,(\sum_{i=0}^{n-1} w_i z_{k,i} + \theta)$, with the synaptic *weights* $w_i \in \mathbb{R}$, *thresholds* $\theta \in \mathbb{R}$, and *sgn* the sign function. Two cost functions commonly associated to a NN are: (*i*) *depth*, which is the number of layers (or the number of edges—if we consider unit length for all the edges connecting the TGs) on the longest input to output path; and (*ii*) *size* (or node complexity), which is the number of neurons (TGs). Unfortunately, these measures are not the best criteria for ranking different solutions when going for silicon [21], as *"comparing the number of nodes is inadequate for comparing the complexity of neural networks as the nodes themselves could implement quite complex functions"* [41].

The fact that *size* is not a good estimate for *area* is explained as: (*i*) the *area* (of one neuron) can be related to its associated *weights*; and (*ii*) the *area* of the connections is—in most cases—neglected. Here are several alternatives of how the *area* could scale in relation to *weights* and *thresholds*:

- for purely digital implementation, the *area* scales at least with the cumulative size of the *weights* and *thresholds* (the bits for representing these *weights* and *thresholds* have to be stored);
- for certain types of analog implementations (*e.g.*, using resistors or capacitors), the same type of scaling

is valid (in particular cases, analog implementations can have binary encoding, thus the *area* would scale with the cumulative *log*-scale size of the parameters);
- there are some types of implementations (*e.g.*, transconductance ones) which offer a constant size per element, thus scaling only with Σ_{NN}*fan-ins*.

All these 'cost functions' are linked to VLSI by the assumptions one makes on how the *area* of a chip scales with the *weights* and the *thresholds* [5, 9, 10]. That is why several other measures (*i.e.*, 'cost functions')—beside *size*—have already been used:

- the total *number-of-connections*, or Σ_{NN}*fan-ins*, has been used by several authors [1, 25, 30, 33];
- the total *number-of-bits* needed to represent the *weights* and *thresholds* $\Sigma_{NN}(\Sigma_i \lceil \log|w_i|\rceil + \lceil \log|\theta|\rceil)$ [1] has been used by others [22, 23, 41];
- the sum of all the absolute values of the *weights* and *thresholds* $\Sigma_{NN}(\Sigma_i |w_i| + |\theta|)$ has also been advocated [5, 10, 18, 19, 21], while another similar 'cost function' is $\Sigma_{NN}(\Sigma_i w_i^2 + \theta^2)$, which has been used in the context of genetic programming for reaching minimal NNs [44].

The sum of all the absolute values of the *weights* and *thresholds* has also been used as an optimum criterion for: (*i*) linear programming synthesis [32]; (*ii*) defining the minimum-integer TG realisation of a function [27]. Recently [3], the same measure (under the name of *"total weight magnitude"*) has been used in the context of computational learning theory. It was proven that the generalisation error of NNs used for classification depends on the size of the *weights*—rather than the number of *weights*—by showing that the misclassification probability converges at a rate of $O\{(cA)^{l}/\sqrt{m}\}$. Here A is the sum of the magnitude of the *weights*, l is the *depth*, m is the number of examples, and c is a constant.

With respect to *delay*, two VLSI models have been commonly in use [38]:

- the simplest one assumes that *delay* is proportional to the input capacitance, hence a TG introduces a *delay* proportional to its *fan-in*;
- a more exact one considers the capacitance along any wire, hence the *delay* is proportional to the *length* of the connecting wires.

It is worth emphasising that it is anyhow desirable to limit the range of parameter values [42] for VLSI implementations—be they digital or analog—because both the maximum value of the *fan-in* [28, 40] and the maximal ratio between the largest and the smallest *weight* [23, 24, 29, 42] cannot grow over a certain (technological) limit.

[1] In this paper $\lfloor x \rfloor$ is the floor of x, *i.e.* the largest integer less than or equal to x, and $\lceil x \rceil$ is the ceiling of x, *i.e.* the smallest integer greater or equal to x. In this paper all the logarithms are to the base 2, except explicitly mentioned otherwise.

The focus of this paper will be on NNs having limited *fan-in* (the *fan-in* will be denoted by Δ). We will present both theoretical proofs and simulations in support of our claim that VLSI- and *size*-optimal NNs can be obtained for small *fan-ins*. For simplification, we shall consider only NNs having n binary inputs and μ binary outputs (if real inputs and outputs are needed, it is always possible to quantize them up to a certain number of bits such as to achieve a desired precision [4, 7, 11, 24]). In Section 2 we shall present a theoretical proof for the $\mathbb{F}_{n,m}$ class of functions, showing that their VLSI-optimal implementation is achieved with small constant *fan-ins*. Section 3 deals with BFs, and details the generalisation of a result from [26] to arbitrary *fan-ins*. Based on that generalisation we will show that the *size* can be minimised for small *fan-ins*. Finally, in Section 4 we will suggest how to implement $\mathbb{F}_{n,m}$ functions in *size*-optimal NNs having small constant *fan-ins*.

Conclusions, open questions and further directions for research complete the paper. Due to space limitations some of the lengthy mathematical proofs suggested in [6, 7] have been omitted, but can be found in [9, 10].

2. VLSI-optimal neural implementations of $\mathbb{F}_{n,m}$ functions

$\mathbb{F}_{n,m}$ is the class of BFs of n variables having m groups of ones in their truth table. Obviously, any BF can be represented by a suitable collection of its true values (ones), but for achieving that the number of groups of ones grows exponentially (*i.e.* $\mathbb{F}_{n,2^{n/2}}$ completely covers $\mathbb{B}_n$, the set of all n-ary BFs). This class of functions has been introduced and analysed by Red'kin [34]; he constructively proved that a *depth*-3 NN has *size* .

Proposition 1 (Theorem 1 from [34]) *The complexity realisation (i.e., number of threshold elements) of* $\mathbb{F}_{n,m}$ *(the class of Boolean functions* $f(x_1 x_2 \ldots \underline{x_{n-1}} x_n)$ *that have exactly m groups of ones) is at most* $2\sqrt{2m}+3$.

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

Red'kin also proved that if the implementation of BFs of this type is restricted to circuits having no more than three layers, than the upper bound—following his method of synthesis—is equal to the lower bound obtained from capacity considerations. Although this construction is *size*-optimal, it is not VLSI-optimal as the *weights* and *thresholds* are exponential.

Another solution was detailed in [20, 21] and later improved in [5, 15, 16]. It has a first layer of COMPARISONs followed by a second layer of MAJORITY gates. This solution relies on the classical COMPARISON of two binary numbers $X = x_{n-1}x_{n-2}\ldots x_1x_0$ and $Y = y_{n-1}y_{n-2}\ldots y_1y_0$, which is a BF defined as:

$$C_n^{>(\geq)} = C_n^{>(\geq)}(X,Y) = \begin{cases} 1 & \text{if } X > Y \ (X \geq Y) \\ 0 & \text{if } X \leq Y \ (X < Y) \end{cases}.$$

It is known from previous work that COMPARISON cannot be computed by a single TG with polynomially bounded integer *weights*, but can be computed by a *depth*-2 NN with $O(n^4)$ TGs and polynomially bounded *weights* [2]. This last result has been improved as follows.

Proposition 2 (Lemma 6 [37]) *The COMPARISON function can be computed in a depth-3 neural network of size 3n with polynomially bounded integer weights.*

This constructive solution (we shall call it **SRK**) has a first layer of n AND gates computing $x_i \wedge \bar{y}_i$, and n OR gates computing $x_i \vee \bar{y}_i$, followed by a layer of $n-1$ AND gates:

$$B_k = (x_k \wedge \bar{y}_k) \wedge \{\wedge_{j=k+1}^{n-1}(x_j \vee \bar{y}_j)\},$$

and a third layer having one OR gate:

$$C_n^{\geq} = (x_{n-1} \wedge \bar{y}_{n-1}) \vee (\vee_{k=0}^{n-2} B_k).$$

This *depth*-3 NN has $size_{\text{SRK}} = 3n-1$, *thresholds* $\leq n$, *fan-in* $\leq n$, and all the *weights* ± 1.

Proposition 3 (Proposition 2 and 3 [7]) *The COMPARISON of two n-bit numbers can be computed by a Δ-ary tree neural network having integer weights and thresholds which are: (i) polynomially bounded for all the values of the fan-in* $3 \leq \Delta \leq O(\log n)$*; (ii) super-polynomially bounded for all the values of the fan-in* $O(\log n) < \Delta \leq O(\log^k n)$*; and (iii) exponentially bounded for all the values of the fan-in* $O(\log^k n) < \Delta \leq 2n$.

This constructive class of solutions (which we shall call B_Δ), was proposed in [18, 19], and is based on decomposing COMPARISON in a tree structure. The NN has a first layer of 'partial' COMPARISONs $C_\Delta^>$ and $C_\Delta^\geq$ ($\lfloor \Delta/2 \rfloor$ bits from X and $\lfloor \Delta/2 \rfloor$ bits from Y) followed by a Δ-ary tree of TGs combining these partial results. The fact that the BFs implemented by the nodes are linear separable functions was proven in [17, 21]. The network has:

$$depth_{\text{B\_}\Delta} = \lceil \log n / (\log\Delta - 1) \rceil, \text{ and}$$

$$size_{\text{B\_}\Delta} = \lceil 4(n-1)/\Delta - 2 \rceil - depth_{\text{B\_}\Delta}$$

with *fan-in* of Δ or Δ − 1. The *weights* and the *thresholds* are lower than $2^{\Delta/2}$ (for $3 \leq \Delta \leq 2n$).

Proposition 4 (Theorem 3 [35]) *The size complexity of COMPARISON implemented by generalised symmetric functions is* $\Theta(n/\log n)$.

This constructive solution (we shall call it **ROS**) has a first layer of 'partial' COMPARISONs C_i (equivalent to $C_{2m}^>$) and $\tilde{C}_i$ (equivalent to $C_{2m}^\geq$) having m input bits from X and m input bits from Y. The first layer has $2\lceil n/m \rceil - 1$ TGs of *fan-in* $= 2m$. The second layer has $\lceil n/m \rceil - 1$ AND gates with *fan-in* $= 2, 3, \ldots, \lceil n/m \rceil$:

$$B_k = \tilde{C}_k \wedge (\wedge_{j=k+1}^{\lceil n/m \rceil} C_j).$$

The third layer has one OR gate $C_n^\geq(X,Y) = \vee_{k=1}^{\lceil n/m \rceil} B_k$, and by taking $m = \lceil \log n \rceil + 1$, $size_{\text{ROS}} = 3\lceil n/(\lceil \log n \rceil + 1) \rceil - 1$. The NN has *depth*-3, *weights* and *thresholds* $\leq 2^{\lceil \log n \rceil}$ and *fan-in* $\leq \lceil n/(\lceil \log n \rceil + 1) \rceil$.

Proposition 5 (Corollary 2 [39]) *The COMPARISON can be computed by a depth-2 linear threshold network of size* $2\lceil n/\lceil\sqrt{n}\rceil\rceil$*, with weight values at most* $2^{\lceil\sqrt{n}\rceil}$ *and with an upper bound of* $2\lceil\sqrt{n}\rceil + 1$ *for the maximum fan-in.*

This constructive solution (we shall call it **VCB**) has a first layer of 'partial' COMPARISONs ($\lceil\sqrt{n}\rceil$ input bits from X and $\lceil\sqrt{n}\rceil$ input bits from Y), and the second layer computes the carry-out of the 2–1 binary ADDITION with carry.

Table 1 (from [7])
***Size*, *weights* and *fan-ins* of different solutions for COMPARISON (normal operand lengths).**

Length	Solution	VCB [39]	B_2⌈√n⌉ [6]	SRK [37]	ROS [35]	VCB [39]	B_Δ [18, 19]	B_Δ [18, 19]
		depth = 2		*depth* = 3				*depth* = 4
32–bits	*Size*	13	**12**	95	19	**18**	19	**45**
	Max. *weight*	64	**32**	32	64	16	**8**	**3**
	Max. *fan-in*	13	**12**	32	12	9	**8**	**5**
64–bits	*Size*	17	**16**	191	31	**26**	39	**63**
	Max. *weight*	256	**128**	64	128	64	**8**	**5**
	Max. *fan-in*	17	**16**	64	14	13	**8**	**7**

Table 2

Different estimates for the AT^2 of COMPARISON for SRK, B_4, B_log, ROS and VCB.

Delay / Area	*Depth*	*Fan-in*	*Length*
Size	$AT^2_{\mathrm{VCB}} = O(\sqrt{n})$	$AT^2_{\mathrm{B\_4}} = O(n\log^2 n)$ (2)	$AT^2_{\mathrm{VCB}} = O(n^2\sqrt{n})$
	$AT^2_{\mathrm{ROS}} = O(n/\log n)$	$AT^2_{\mathrm{B\_log}} = O[n\log^3 n/\log^2(\log n)]$	$AT^2_{\mathrm{ROS}} \cong 3\cdot n^3/\log n$
	$AT^2_{\mathrm{SRK}} = O(n)$	$AT^2_{\mathrm{VCB}} = O(n\sqrt{n})$	$AT^2_{\mathrm{B\_log}} \cong 4\cdot n^3/\log n$
	$AT^2_{\mathrm{B\_log}} = O[n\log n/\log^2(\log n)]$	$AT^2_{\mathrm{ROS}} = O(n^3/\log^3 n)$	$AT^2_{\mathrm{B\_4}} \cong 4\cdot n^3$
	$AT^2_{\mathrm{B\_4}} = O(n\log^2 n)$	$AT^2_{\mathrm{SRK}} = O(n^3)$	$AT^2_{\mathrm{SRK}} \cong 27n^3/4$
$\sum_{NN}$ *fan-ins*	$AT^2_{\mathrm{VCB}} = O(n)$	$AT^2_{\mathrm{B\_4}} = O(n\log^2 n)$ (3)	$AT^2_{\mathrm{B\_log}} \cong 4n^3$
	$AT^2_{\mathrm{B\_log}} = O[n\log^2 n/\log^2(\log n)]$	$AT^2_{\mathrm{B\_log}} = O[n\log^4 n/\log^2(\log n)]$	$AT^2_{\mathrm{VCB}} \cong 4n^3$
	$AT^2_{\mathrm{B\_4}} = O(n\log^2 n)$	$AT^2_{\mathrm{VCB}} = O(n^2)$	$AT^2_{\mathrm{B\_4}} \cong 5n^3$
	$AT^2_{\mathrm{ROS}} = O(n^2/\log^2 n)$	$AT^2_{\mathrm{ROS}} = O(n^4/\log^4 n)$	$AT^2_{\mathrm{ROS}} = O(n^4/\log^2 n)$
	$AT^2_{\mathrm{SRK}} = O(n^2)$	$AT^2_{\mathrm{SRK}} = O(n^4)$	$AT^2_{\mathrm{SRK}} = O(n^4)$
$\sum_{NN}(\sum_i \lvert w_i\rvert + \lvert\theta\rvert)$	$AT^2_{\mathrm{B\_4}} = O(n\log^2 n)$ (1)	$AT^2_{\mathrm{B\_4}} = O(n\log^2 n)$ (4)	$AT^2_{\mathrm{B\_4}} = O(n^3)$
	$AT^2_{\mathrm{B\_log}} = O[n\sqrt{n}\log n/\log^2(\log n)]$	$AT^2_{\mathrm{B\_log}} = O[n\sqrt{n}\log^3 n/\log^2(\log n)]$	$AT^2_{\mathrm{B\_log}} = O(n^3\sqrt{n}/\log n)$
	$AT^2_{\mathrm{ROS}} = O(n^2/\log n)$	$AT^2_{\mathrm{ROS}} = O(n^4/\log^3 n)$	$AT^2_{\mathrm{ROS}} = O(n^4/\log n)$
	$AT^2_{\mathrm{SRK}} = O(n^2)$	$AT^2_{\mathrm{SRK}} = O(n^4)$	$AT^2_{\mathrm{SRK}} = O(n^4)$
	$AT^2_{\mathrm{VCB}} = O(n^{1/2}2^{\sqrt{n}})$	$AT^2_{\mathrm{VCB}} = O(n^{3/2}2^{\sqrt{n}})$	$AT^2_{\mathrm{VCB}} = O(n^{5/2}2^{\sqrt{n}})$

Although *size*-optimal (like [34]), this solution is not very attractive from the VLSI complexity point of view as having exponentially growing *weights*. The complexity results are similar to those of a particular solution belonging to the B_Δ class [6, 7]: take $\Delta = 2\lceil\sqrt{n}\,\rceil$—which leads to B_2$\lceil\sqrt{n}\,\rceil$—and the NN has *depth* = 2, *size* = $\lceil 2\sqrt{n}\,\rceil$, with *weights* and *thresholds* of at most $2^{\lceil\sqrt{n}\,\rceil}$.

For normal length COMPARISONs Vassiliadis *et al.* [39] claim improvements over **ROS** [35] and **SRK** [37]. We present in *Table 1* the results reported in [7]. Both **VCB** and B_2$\lceil\sqrt{n}\,\rceil$ achieve better performances than **SRK** and **ROS**. For *depth* = 2, B_2$\lceil\sqrt{n}\,\rceil$ outperforms **VCB** (both for 32-bit and for 64-bit operand lengths), while for *depth* = 3, B_2$\lceil\sqrt{n}\,\rceil$ has lower *weights* and *fan-ins*, but slightly more gates. Still, B_Δ has two main advantages: (*i*) being a class of solutions it can be used for other *depths* (see the last column of *Table 1*); (*ii*) because the *weights* and the *fan-ins* are lower, the *area* should also be lower.

It is known that a VLSI design is considered optimum when judged on a combined measure AT^2 [38], thus:

$$AT^2_{\mathrm{SRK}} = (3n-1)\times 3^2 \qquad = O(n)$$

$$AT^2_{\mathrm{B\_\Delta}} < \left\lceil\frac{4(n-1)}{\Delta-2}\right\rceil \times \left\lceil\frac{\log n}{\log\Delta - 1}\right\rceil^2 = O(n\log^2 n/\Delta\log\Delta)$$

$$AT^2_{\mathrm{ROS}} = 3\lceil n/(\lceil\log n\rceil + 1)\rceil \times 3^2 = O(n/\log n).$$

The natural extension of the circuit complexity results to VLSI complexity ones is by using the closer estimates for the *area* and the *delay* (as discussed in *Introduction*).

Proposition 6 *If the area of a neural network is estimated as* $\sum_{NN}$ *fan-ins, there are neural networks computing the* COMPARISON *of two n-bit numbers which occupy between* $O(n)$ *and* $O(n^2)$ *area:*

$$A_{\mathrm{SRK}} = (n^2 + 11n - 6)/2 \qquad = O(n^2)$$

$$A_{\mathrm{B\_\Delta}} < \Delta\times\left\lceil\frac{4(n-1)}{\Delta-2}\right\rceil + \left\lceil\frac{\log n}{\log\Delta - 1}\right\rceil + \left\lceil\frac{4n}{\Delta}\right\rceil = O(n)$$

$$A_{\mathrm{ROS}} < 4n + \frac{1}{2}\cdot\left\lceil\frac{n}{\lceil\log n\rceil + 1}\right\rceil^2 + \frac{3}{2}\cdot\left\lceil\frac{n}{\lceil\log n\rceil + 1}\right\rceil$$

$$= O(n^2/\log^2 n).$$

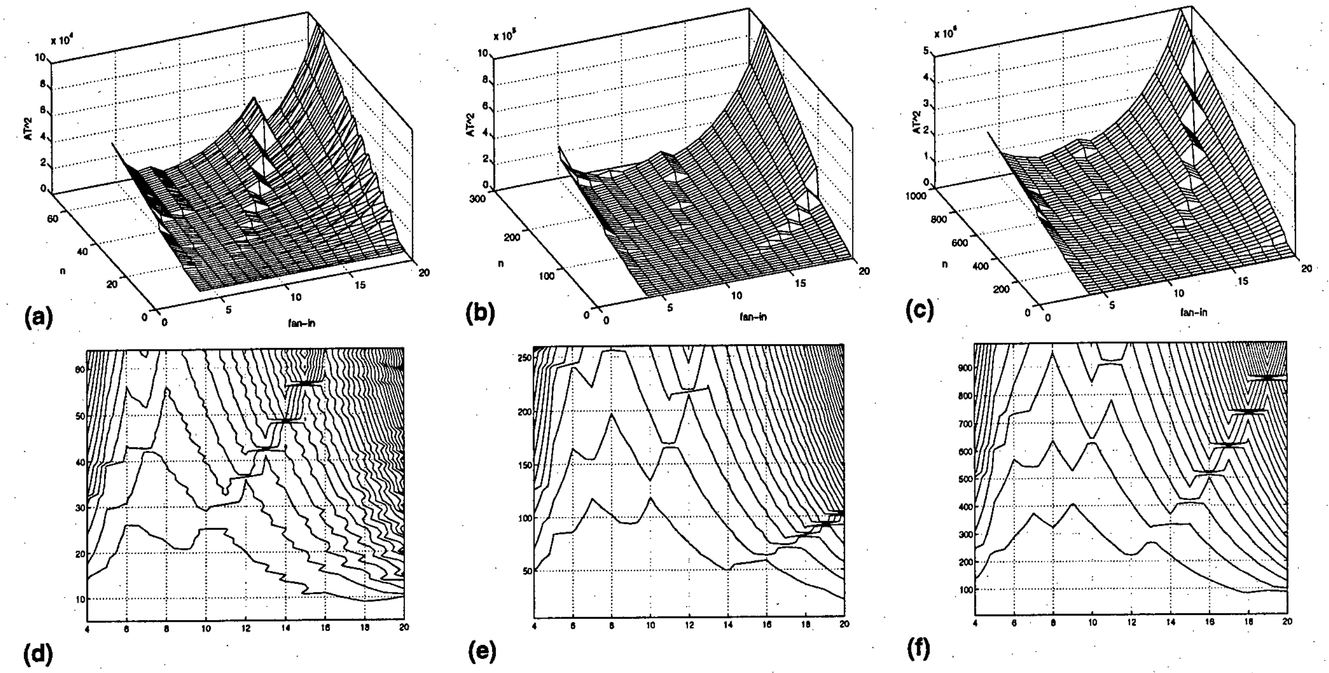

Figure 1. The AT^2 values of COMPARISON—plotted as a 3D surface—versus the number of inputs n and the *fan-in* Δ for $4 \le \Delta \le 20$: (a) $n \le 64$; (b) $n \le 256$; (c) $n \le 1024$; (d), (e), and (f) show the contour plots for the same cases. Clearly a 'valley' is formed, and that the 'deepest' points constantly lie somewhere between $\Delta_{minim} = 6$ and $\Delta_{maxim} = 9$.

Proposition 7 *If the area of a neural network is estimated as $\Sigma_{NN}(\Sigma_i |w_i| + |\theta|)$, there are neural networks computing the* COMPARISON *of two n-bit numbers which occupy between* $O(n)$ *and* $O(n^2)$ *area:*

$$A_{SRK} = (2n^2 + 15n - 9)/2 \quad = O(n^2)$$

$$A_{B\_\Delta} < \frac{2^{\Delta/2}}{\Delta} \cdot \frac{8n\Delta - 6n - 5\Delta}{\Delta - 2} \quad = O(n \cdot 2^{\Delta/2}/\Delta)$$

$$A_{ROS} < \left(4n - \frac{5}{2}\right) \times \left\lceil \frac{n}{\lceil \log n \rceil + 1} \right\rceil + \left\lceil \frac{n}{\lceil \log n \rceil + 1} \right\rceil^2 = O(n^2/\log n).$$

Proposition 8 *If the delay of one neuron is proportional to its fan-in, there are neural networks computing the* COMPARISON *of two n-bit numbers which require between* $O(\log n)$ *and* $O(n)$ *time:*

$$T_{SRK} = 2n + 2 \quad = O(n)$$

$$T_{B\_\Delta} < (\Delta - 1) \times \left\lceil \frac{\log n}{\log\Delta - 1} \right\rceil + 1 \quad = O(\Delta \log n / \log\Delta)$$

$$T_{ROS} < 2 \times \left(\left\lceil \frac{n}{\lceil \log n \rceil + 1} \right\rceil + \lceil \log n \rceil + 1 \right) = O(n/\log n).$$

Proposition 9 *If the delay in a neural network is proportional to the length of the wires, there are neural networks computing the* COMPARISON *of two n-bit numbers which require* $O(n)$ *time:*

$$T_{SRK} = (3n - 1)/2, \qquad T_{B\_\Delta} < n, \qquad T_{ROS} < n.$$

For all these different estimations of A and T, the AT^2 complexity values have been computed, ordered, and can be seen in *Table 2*.

Not wanting to complicate the proof, we shall determine the VLSI-optimal *fan-in* when implementing COMPARISON (in fact an $\mathbb{F}_{n,1}$ function) for which several solutions were detailed in *Propositions* 2 to *9*. The same result is valid for $\mathbb{F}_{n,m}$ functions as can be intuitively expected as:

- the delay is determined by the first layer of COMPARISONs; while
- the *area* is mostly influenced by the same first layer of COMPARISONs (the *area* for the implementing the MAJORITY gate can be neglected [15, 21]).

From the alternatives presented in *Table 2*, we have chosen $\Sigma_{NN}(\Sigma_i |w_i| + |\theta|)$ for *area* and *depth* for delay, but other estimates lead to similar results (the optimal AT^2 being $O(n \log^2 n)$ in four out of nine cases—see *Table 2*). To get a better understanding, the AT^2 values have been computed for variable *fan-ins* and for different number of inputs n, and can be seen in *Figure 1*.

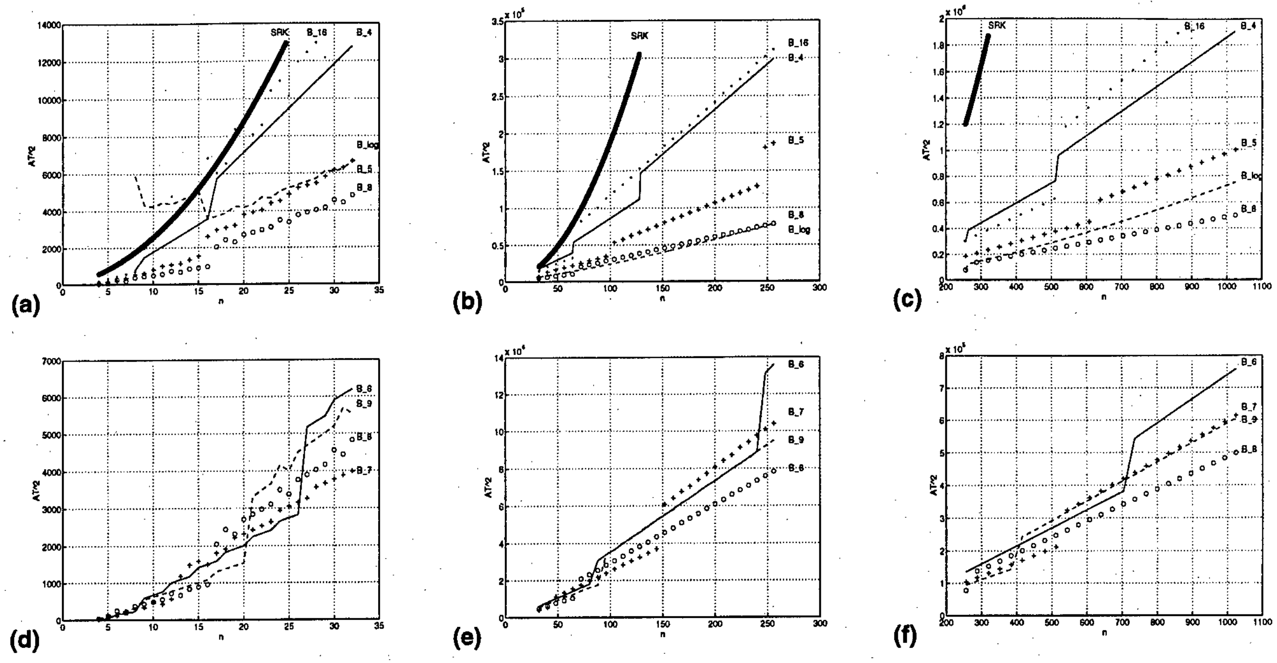

Figure 2. The AT^2 values of COMPARISON for different number of inputs n and *fan-in* Δ for the intervals: (a) $4 \le n \le 32$; (b) $32 \le n \le 256$; and (c) $256 \le n \le 1024$. Details showing the 'optimum' *fan-in* for the same intervals: (d) $4 \le n \le 32$; (e) $32 \le n \le 256$; and (f) $256 \le n \le 1024$.

Proposition 10 *The VLSI-optimal neural network which computes the* COMPARISON *of two n-bit numbers has small-constant fan-in 'neurons' with small-constant bounded weights and thresholds.*

Proof From *Propositions 3* and *7*:

$$AT^2 \equiv \frac{2^{\Delta/2}}{\Delta}\cdot\frac{8n\Delta - 6n - 5\Delta}{\Delta - 2}\cdot\left(\frac{\log n}{\log\Delta}\right)^2$$

$$= O\{n\log^2 n \cdot 2^{\Delta/2}/(\Delta\log^2\Delta)\} \qquad (1)$$

and we compute the derivative:

$$\frac{d(AT^2)}{d\Delta} = \frac{2^{\Delta/2}\log^2 n}{\Delta^2(\Delta-2)^2\log^3\Delta}\times\Big(8n\Delta^3\log\Delta - 22n\Delta^2\log\Delta + 12n\Delta\log\Delta - 5\Delta^3\log\Delta + 10\Delta^2\log\Delta - \frac{16}{\ln 2}n\Delta^2\log\Delta + \frac{24}{\ln 2}n\Delta\log\Delta - \frac{24}{\ln 2}n\log\Delta + \frac{10}{\ln 2}\Delta^2\log\Delta - \frac{32}{\ln 2}n\Delta^2 + \frac{88}{\ln 2}n\Delta - \frac{48}{\ln 2}n + \frac{20}{\ln 2}\Delta^2 - \frac{40}{\ln 2}\Delta\Big).$$

This—unfortunately—involves transcendental functions of the variables in an essentially non-algebraic way. By considering the simplified 'complexity' version (1) we have:

$$d(AT^2)/d\Delta \equiv d\{n\log^2 n\cdot 2^{\Delta/2}/(\Delta\log^2\Delta)\}/d\Delta$$

$$= \frac{2^{\Delta/2}}{\Delta\log^2\Delta}\cdot\left(\frac{\ln 2}{2} - \frac{1}{\Delta} - \frac{2}{\Delta\ln\Delta}\right),$$

which when equated to zero leads to $\ln\Delta\,(\Delta\ln 2 - 2) = 4$ (again a transcendental equation). This has $\Delta_{optim} = 6$ as integer solution, and because the *weights* and the *thresholds* are bounded by $2^{\Delta/2}$ (*Proposition 3*) the proof is concluded. ❑

To get a better understanding, the AT^2 values have been computed for variable *fan-ins* and different number of inputs n, as can be seen in *Figure 1*, while *Figure 2* presents exact plots of the AT^2 measure which support our previous claim $\Delta_{optim} = 6\ldots9$ (as the proof has been obtained using several approximations: neglecting ceilings, using the complexity estimate, etc.).

3. Size-optimal neural implementations of BFs

We start from the classical construction developed by Shannon [36] for synthesising one BF with *fan-in* 2 AND-OR gates. It was extended to the multioutput case and modified to apply to NNs by Horne & Hush [26].

Proposition 11 (Theorem 3 [26]) *Arbitrary Boolean logic functions* $f:\{0,1\}^n \rightarrow \{0,1\}^\mu$ *can be implemented in a neural network of perceptrons restricted to fan-in 2 with a node complexity of* $\Theta\{\mu 2^n/(n+\log\mu)\}$. *The resulting architecture requires **O** (n) layers.*

Sketch of proof The idea is to decompose each output BF into two subfunctions using Shannon's Decomposition [36]:

$$f(x_1 x_2 \ldots x_{n-1} x_n) = \bar{x}_1 f_0(x_2 \ldots x_{n-1} x_n) + x_1 f_1(x_2 \ldots x_{n-1} x_n).$$

By doing this recursively for each subfunction, the output BFs will—in the end—be implemented by binary trees. Horne & Hush [26] 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:

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

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:

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

nodes and $depth_{lower} = 2q$ (see Figure 2 from [26]). That q which minimises the *size* of the two subnetworks is determined by solving $d\,(size)/dq = 0$, and gives:

$$q \approx \log\,[n + \log\mu - 2\log(n+\log\mu)]\,.$$

By substituting this value in (2) and (3), the minimum *size*:

$$size \cong 3\mu \cdot 2^{n-q} \cong 3\mu \cdot 2^n/(n+\log\mu)$$

is obtained. ❑

We will use a similar approach for the case when the *fan-in* is limited by Δ.

Proposition 12 *Arbitrary Boolean functions* $f:\{0,1\}^n \rightarrow \{0,1\}^\mu$ *can be implemented in a neural network of perceptrons restricted to fan-in* Δ *in **O*** $(n/\log\Delta)$ *layers.*

Proof We use the approach of Horne & Hush [26] and limit the *fan-in* 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 first decomposition step eliminates $\Delta - 1$ variables and generates a tree of:

$$depth = 1 + \lceil(\Delta-1)/\log\Delta\rceil,$$

$$size = 2^{\Delta-1} + \lceil(2^{\Delta-1}-1)/(\Delta-1)\rceil.$$

Repeating this procedure recursively k times, we have:

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

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

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:

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

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

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 $\alpha + 1$:

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

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

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

(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:

$$depth_{BFs} = \{k + \lfloor (n - k\Delta)/\Delta \rfloor\} \cdot \{1 + \lceil (\Delta - 1)/\log\Delta \rceil\}$$
$$= (n/\Delta) \cdot (\Delta/\log\Delta + 1) \qquad (9)$$
$$\approx n/\log\Delta = O(n/\log\Delta)$$
$$size_{BFs} = \mu \cdot size \cdot \{2^{k(\Delta-1)} - 1\}/(\Delta - 1) + (size + 1) \cdot 2^{2^{n-(k+1)\Delta}}$$
$$\approx \mu \cdot 2^{k\Delta - k} + 2^{\Delta} \cdot 2^{2^{n-k\Delta-\Delta}} \qquad (10)$$

concluding the proof. ❑

Proposition 13 *All the critical points of $size_{BFs}(\mu, n, k, \Delta)$ are relative minimum and are situated in the (close) vicinity of the parabola $k\Delta \approx n - \log(n + \log\mu)$.*

Proof To determine the critical points, we equate the partial derivatives to zero. Starting from the approximation (10) of $size_{BFs}$ we compute $\partial size_{BFs}/\partial k = 0$:

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

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

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

which has an approximate solution $\gamma \approx n - \log(n + \log\mu)$.

The same result can be obtained by computing with finite differences (instead of approximating the partial derivative):

$$size_{BFs}(\mu, n, k+1, \Delta) - size_{BFs}(\mu, n, k, \Delta) = 0$$
$$size \cdot \{\mu \cdot 2^{k\Delta - k} - 2^{2^{n-k\Delta-\Delta}}\} = 0$$
$$\mu \cdot 2^{k\Delta - k} = 2^{2^{n-k\Delta-\Delta}}$$

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

$$\log\{\log\mu + \gamma(1 - 1/\Delta)\} = n - \gamma - \Delta$$
$$\gamma = n - \{\Delta + \log(1 - 1/\Delta)\} - \log\{\gamma + \Delta/(\Delta-1) \cdot \log\mu\}$$
$$\approx n - \Delta - \log(\gamma + \log\mu), \qquad (12)$$

which has as the approximate solution:

$$\gamma = n - \log(n + \log\mu).$$

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

$$\mu\, 2^{k\Delta - k}(\ln 2)\, k + 2^{\Delta}(\ln 2)\, 2^{2^{n-k\Delta-\Delta}} + 2^{\Delta} 2^{2^{n-k\Delta-\Delta}}(\ln 2)\, 2^{n-k\Delta-\Delta}(\ln 2)(-k) = 0$$
$$\mu k \cdot 2^{\gamma - k} = k(\ln 2) \cdot 2^{n-\gamma} \cdot 2^{2^{n-\gamma-\Delta}} - 2^{\Delta} \cdot 2^{2^{n-\gamma-\Delta}}$$
$$\mu k \cdot 2^{\gamma - k} \cdot 2^{\gamma - n} = k(\ln 2) \cdot 2^{2^{n-\gamma-\Delta}} - 2^{\Delta} \cdot 2^{\gamma - n} \cdot 2^{2^{n-\gamma-\Delta}}$$
$$\mu k \cdot 2^{2\gamma - k - n} = \{k(\ln 2) - 2^{\gamma + \Delta - n}\} \cdot 2^{2^{n-\gamma-\Delta}}$$
$$(\mu/\ln 2) \cdot 2^{2\gamma - k - n} = \{1 - 2^{\gamma + \Delta - n}/(k \ln 2)\} \cdot 2^{2^{n-\gamma-\Delta}}$$

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 $k\Delta \approx n - \log(n + \log\mu)$. ❑

From *Proposition 12* and *13* it follows that *size*-optimal neural implementations of BFs are obtained for small *fan-ins* (*i.e.*, from constant to at most $n - \log n < n$). The exact *size*:

$$size_{BFs} = size_{lower} + \mu \cdot size_{upper}$$

has been computed for many different values of n, μ, Δ and k. Some results of those extensive simulations are plotted in *Figure 3*. From *Figure 3(a), (b)* and *(c)* it may seem that k and Δ used in the proof of *Proposition 12* have the same influence on $size_{BFs}$. The discrete parabola-like curves approximating $k\Delta \approx n - \log(n + \log\mu)$ can be seen in *Figure 3(d), (e)* and (f).

Proposition 14 *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):

$$size^{*}_{BFs} \approx \mu \cdot 2^{n - \log n - k} + 2^{\Delta} \cdot 2^{2^{n-n+\log n-\Delta}}$$
$$< \mu \cdot 2^{n - \log n} + 2^{\Delta} \cdot 2^{2^{\log n}}$$
$$= \mu \cdot 2^{n}/n + 2^{\Delta} \cdot 2^{n},$$

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

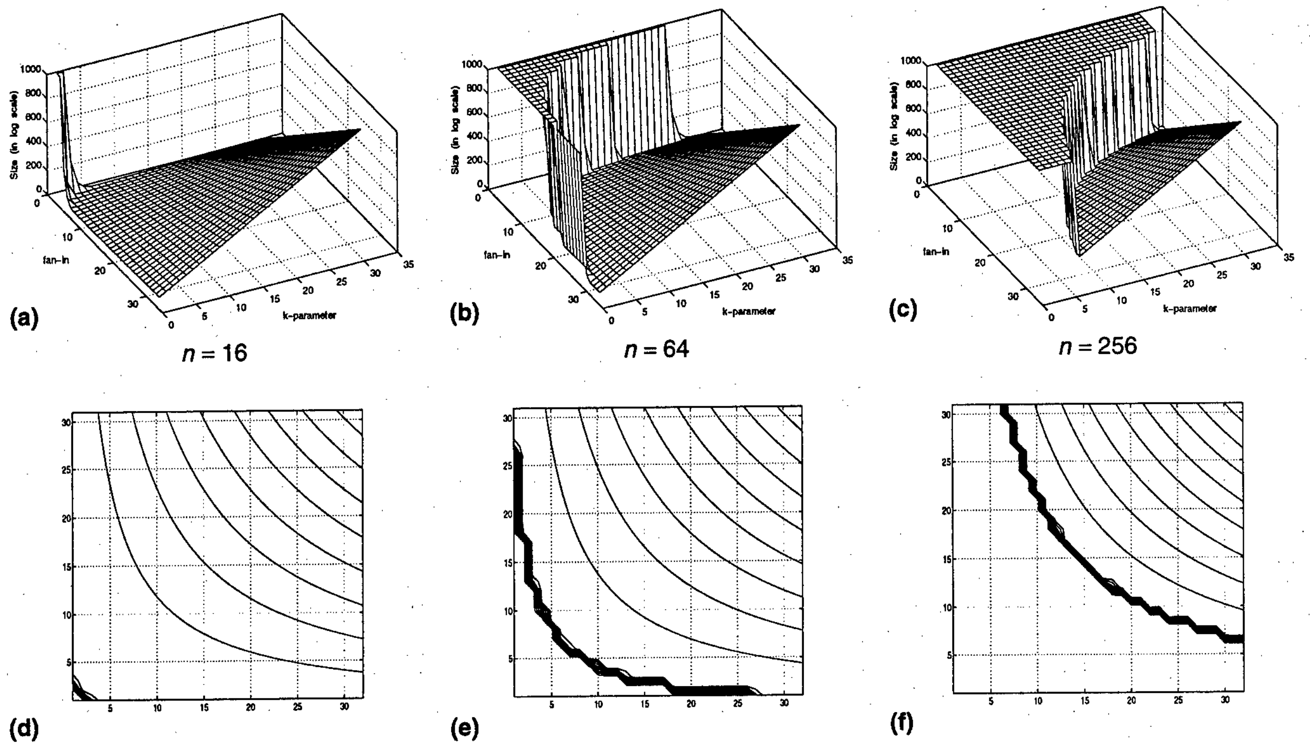

Figure 3. The *size* (in logarithmic scale) of NNs implementing arbitrary BFs for: (a) $n = 16$; (b) $n = 64$; (c) $n = 256$ (clipped at 2^{1000}), and the contour plots for the same cases (d), (e), (f).

The detailed proof computes $size_{BFs}(n, \mu, k, \Delta)$ for those $k \approx (n - \log n)/\Delta$ (*i.e.*, $size^*_{BFs}(n, \mu, \Delta)$), and shows that:

$$size^*_{BFs}(n, \mu, \Delta + 1) - size^*_{BFs}(n, \mu, \Delta) > 0.$$

Hence, the function is 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 3*. 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 important that the other relative minima (on, or in the vicinity of the parabola $k\Delta \approx n - \log n$) are only 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 mentioned that all these relative minimum are obtained for *fan-ins* strictly lower that linear, as $\Delta \leq n - \log n$.

4. Size-optimal neural implementations of $\mathbb{F}_{n,m}$ functions

A similar result can be obtained for $\mathbb{F}_{n,m}$, as the first layer is represented by COMPARISONs (*i.e.*, $\mathbb{F}_{n,1}$) which can be decomposed to satisfy the limited *fan-in* condition [10, 16, 17, 21].

Table 3
Minimum $size_{BFs}$ for different values of n and $\mu = 1$.

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

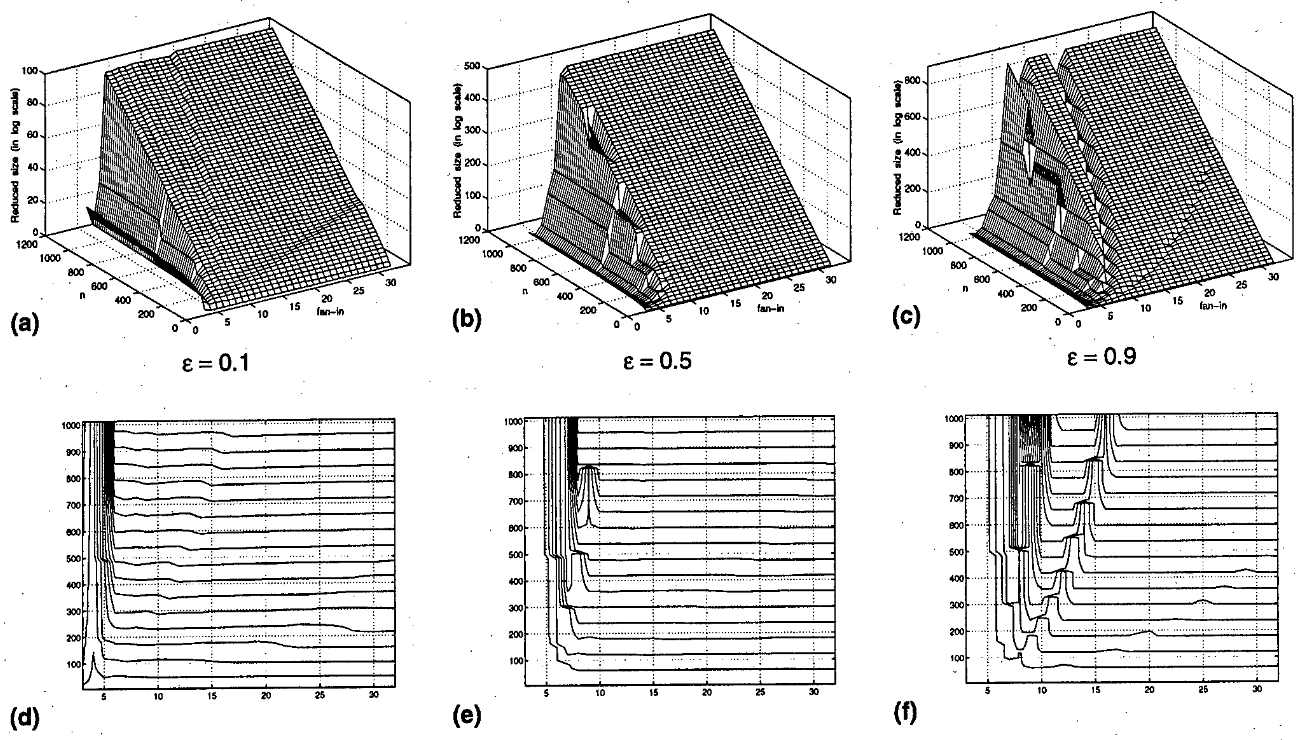

Figure 4. The reduced *size* (in logarithmic scale) of NNs implementing $I\!F_{n,m}$ functions for $m = 2^{\varepsilon n}$: (a) $\varepsilon = 0.1$; (b) $\varepsilon = 0.5$; (c) $\varepsilon = 0.9$; and the contour plots for the same cases (d), (e), (f). The lowest values are obtained for very small constant *fan-in* values.

Proposition 15 (Lemma 1 & Corollary 1 [21]) *The* COMPARISON *of two n-bit numbers can be computed by a Δ-ary tree neural network having integer weights and thresholds bounded by $2^{\Delta/2}$ for any $3 \le \Delta \le n$.*

The *size* complexity of the NN implementing one $I\!F_{n,m}$ function is [21]:

$$size_{B\_\Delta} = 2nm \cdot \left\{ \frac{1}{\Delta/2} + \dots + \frac{1}{(\Delta/2)^{depth_{B\_\Delta}}} \right\}, \qquad (13)$$

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

$$(2n/\Delta) \cdot 2^{\Delta}, \quad \frac{2n/\Delta}{\Delta/2} \cdot 2^{\Delta(\Delta/2)}, \dots, \quad \frac{2n/\Delta}{(\Delta/2)^{depth_{B\_\Delta}-1}} \cdot 2^{\Delta(\Delta/2)^{depth_{B\_\Delta}-1}}. \qquad (14)$$

For large enough m (needed for achieving a certain precision [10, 23, 42]), and/or large enough n, the first terms of the sum (13) will be larger than the equivalent ones from (14). This is equivalent to the trick from [26], as the lower levels will compute ***all the possible functions*** using only limited *fan-in* COMPARISONs. Hence, the optimum *size* becomes:

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

explained as the same BFs are computed redundantly. In terms of *fan-in*, several exponentially decreasing terms will be replaced by double exponential increasing terms.

Following similar steps to the ones used in *Proposition 13*, it is possible to show that the minimum *size* is obtained for $\Delta = 3$. To get a better understanding we have done extensive simulations by considering that $m = 2^{\varepsilon n}$. Some of the results of these simulations can be seen in *Figure 4*. They show that it is always possible to obtain a significant reduction of the *size* by properly choosing a small constant *fan-in*. It is to be mentioned that the *size* reduction is by a huge factor which is of the form $2^{\varepsilon n - c}$ for very small *fan-ins* $\Delta_{optim} = 4 \dots 6$.

5. Conclusions and further work

The paper has focused on sparsely connected NNs, *i.e.* having either (small) constant *fan-ins*, or at most logarithmic in the number of inputs n. Using different cost functions—which are closer estimates than *size* and *depth* for the *area* and the *delay* of a VLSI chip—we have been able to prove that VLSI-optimal implementations of $\mathbb{F}_{n,m}$ functions are obtained by small constant *fan-ins*.

Concerning *size*-optimal solutions, we have shown that:

- arbitrary BFs require small—but not necessarily constant—*fan-ins* (at most $n - \log n$);
- $\mathbb{F}_{n,m}$ functions require small constant *fan-ins*.

Some of these results have already been applied to optimising the VLSI design of a neural constructive algorithm [5, 14, 15]. We are working on a mixed constructive algorithm which—after quantizing the input space as in [4, 11, 12, 13]—could synthesise $\mathbb{F}_{n,m}$ functions, arbitrary BFs, or a mixture of them such as to reduce the *area* of the resulting VLSI chip. This could also have applications to automatic synthesis of mixed analog/digital circuits [8, 16]. An alternative solution currently under investigation [14] is to use such a synthesis step after quantizing the input space as detailed in [24].

Future work should concentrate on:

- linking such results with the 'Occam's razor' [44] and the 'minimum description length' [43];
- finding closer estimates (*i.e.*, other cost functions) for optimal mixed analogue/digital implementations.

The main conclusion is that VLSI-optimal solutions can be obtained for small constant *fan-ins*. We mention here that there are similar small constants relating to our capacity of processing information [31].

References

[1] Y.S. Abu-Mostafa, "Connectivity Versus Entropy," in D.Z. Anderson (ed.): *Neural Info. Proc. Sys.*, Amer. Inst. of Physics, NY, 1-8, 1988.

[2] N. Alon & J. Bruck, "Explicit Construction of Depth-2 Majority Circuits for Comparison and Addition," *Tech. Rep. RJ 8300 (75661)*, IBM Almaden, San Jose, CA, 1991.

[3] P.L. Bartlett, "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.* (ftp:syseng.anu.edu. au/pub/peter/TR96d.ps.Z), Dept. Sys. Eng., Sch. of Info. Sci. and Eng., Australian National Univ., Canberra 0200 Australia, May 1996 (short version in M.C. Mozer, M.I. Jordan & T. Petsche (eds.): *Advances in Neural Info. Proc. Sys. 9*, MIT Press, MA, 1997).

[4] V. Beiu, "Entropy Bounds for Classification Algorithms," *Neural Network World*, **6**(4), 497–505, 1996.

[5] V. Beiu, "Optimal VLSI Implementation of Neural Networks," in J.G. Taylor (ed.): *Neural Networks and Their Applications*, John Wiley, Chichester, 255-276, June 1996.

[6] V. Beiu, "VLSI Complexity of Threshold Gate COMPARISON," *Conf. Rep. Intl. Symp. on Neuro-Fuzzy Sys.* (AT'96, Lausanne, Switzerland), Adv. & Appls. Tech. Inst. (Aati), Ecubens, 161-170, August 1996.

[7] V. Beiu, "New VLSI Complexity Results for Threshold Gate COMPARISON," in T.B. Ludermir (ed.): *Proc. 3rd Brazilian Symp. on Neural Networks* (III SBRN, Recife, Brazil), UFPE-DI, Recife, 251-258, November 1996.

[8] V. Beiu, "Optimization of Circuits Using a Constructive Learning Algorithm," *Tech. Rep. LA-UR-97-851*, Los Alamos Natl. Lab., USA; in A.B. Bulsari & S. Kallio (eds.): *Neural Networks in Engineering Systems* (EANN'97, Stockholm, Sweden), Systems Engineering Association, Turku, Finland, 291-294, June 1997.

[9] V. Beiu, "On the Circuit and VLSI Complexity of Threshold Gate COMPARISON," *Tech. Rep. LA-UR-96-3591*, Los Alamos Natl. Lab., USA, Dec. 1996; to appear in *Neurocomputing*, 1997/8.

[10] V. Beiu, *VLSI Complexity of Discrete Neural Networks*, Gordon and Breach, Newark, NJ, 1998.

[11] V. Beiu & T. De Pauw, "Tight Bounds on the Size of Neural Networks for Classification Problems," *Tech. Rep. LA-UR-97-60*, Los Alamos Natl. Lab., USA; in J. Mira, R. Moreno-Díaz & J. Cabestany (eds.): *Biological and Artificial Computation: From Neuroscience to Technology* (IWANN'97, Lanzarote, Spain), Lecture Notes in Comp. Sci., vol. **1240**, Springer, Berlin, 743-752, June 1997.

[12] V. Beiu & S. Draghici, "Limited Weights Neural Networks: Very Tight Entropy Based Bounds," *Tech. Rep. LA-UR-97-294*, Los Alamos Natl. Lab., USA; in D.W. Pearson (ed.): *Proc. Intl. ICSC Symp. on Soft Computing* (SOCO '97, Nîmes, France), ICSC Academic Press, Canada, 111-118, September 1997.

[13] V. Beiu & H.E. Makaruk, "Computing n-Dimensional Volumes of Complexes: Application to Constructive Entropy Bounds," *Tech. Rep. LA-UR-97-2873*, Los Alamos Natl. Lab., USA; to appear in *Proc. Intl. Symp. on Nonlinear Theory and Its Applications* (NOLTA'97, Hawaii, USA), November 29 - December 3, 1997.

[14] S. Draghici, V. Beiu & I.K. Sethi, "A VLSI Optimal Constructive Algorithm for Classification Problems," *Tech. Rep. LA-UR-97-1609*, Los Alamos Natl. Lab., USA; to appear in C.H. Dagli, M. Akay, O. Ersoy, B. Fernández & A. Smith (eds.): *Smart Engineering System Design: Neural Networks, Fuzzy Logic, and Evolutionary Programming* (ANNIE'97, St. Louis, USA), November 9-12, ASME Press, Fairfield, NJ, 1997.

[15] V. Beiu & J.G. Taylor, "VLSI Optimal Neural Network Learning Algorithm," in D.W. Pearson, N.C. Steele & R.F. Albrecht (eds.): *Artif. Neural Nets & Genetic Algorithms* (ICANNGA'95, Alès, France), Springer-Verlag, Vienna, 61-64, 1995.

[16] V. Beiu & J.G. Taylor, "Direct Synthesis of Neural Networks," *Proc. MicroNeuro'96* (Lausanne, Switzerland), IEEE CS Press, Los Alamitos, CA, 257-264, 1996.

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

[18] V. Beiu, J.A. Peperstraete, J. Vandewalle & R. Lauwereins, "Efficient Decomposition of COMPARISON and Its Applications," in M. Verleysen (ed.): *ESANN'93* (Brussels, Belgium), Dfacto, Brussels, 45-50, 1993.

[19] V. Beiu, J.A. Peperstraete, J. Vandewalle & R. Lauwereins, "COMPARISON and Threshold Gate Decomposition," in D.J. Myers & A.F. Murray (eds.): *MicroNeuro'93* (Edinburgh, UK), UnivEd Tech. Ltd., Edinburgh, 83-90, 1993.

[20] V. Beiu, J.A. Peperstraete, J. Vandewalle & R. Lauwereins, "Learning from Examples and VLSI Implementation of Neural Networks," in R. Trappl (ed.): *Cybernetics & Sys. Res.'94* (EMCSR'94, Vienna, Austria), World Scientific, Singapore, 1767-1774, 1994.

[21] V. Beiu, J.A. Peperstraete, J. Vandewalle & R. Lauwereins, "Area-Time Performances of Some Neural Computations," in P. Borne, T. Fukuda & S.G. Tzafestas (eds.): *SPRANN '94* (Lille, France), GERF EC, Lille, 664-668, 1994.

[22] J. Bruck & J.W. Goodmann, "On the Power of Neural Networks for Solving Hard Problems," in D.Z. Anderson, (ed.): *Neural Info. Proc. Sys.*, Amer. Inst. of Physics, NY, 137-143, 1988; also in *J. Complexity*, **6**, 129-135, 1990.

[23] J.S. Denker & B.S. Wittner, "Network Generality, Training Required and Precision Required," in D.Z. Anderson (ed.): *Neural Info. Proc. Sys.*, Amer. Inst. of Physics, NY, 219-222, 1988.

[24] S. Draghici & I.K. Sethi, "On the Possibilities of the Limited Precision Weights Neural Networks in Classification Problems," in J. Mira, R. Moreno-Díaz & J. Cabestany (eds.): *Biological and Artificial Computation: From Neuroscience to Technology* (IWANN'97, Lanzarote, Spain), Lecture Notes in Comp. Sci., vol. **1240**, Springer, Berlin, 753-762, 1997.

[25] D. Hammerstrom, "The Connectivity Analysis of Simple Association –or– How Many Connections Do You Need," in D.Z. Anderson (ed.): *Neural Info. Proc. Sys.*, Amer. Inst. of Physics, NY, 338-347, 1988.

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

[27] S. Hu, *Threshold Logic*, Univ. California Press, Berkeley, CA, 1965.

[28] H. Klaggers & M. Soegtrop, "Limited Fan-in Random Wired Cascade-Correlation," in D.J. Myers & A.F. Murray (eds.): *MicroNeuro'93* (Edinburgh, UK), UnivEd Tech. Ltd., Edinburgh, 79-82, 1993.

[29] A.V. Krishnamoorthy, R. Paturi, M. Blume, G.D. Linden, L.H. Linden & S.C. Esener, "Hardware Tradeoffs for Boolean Concept Learning," *Proc. WCNN'94* (San Diego, USA), Lawrence Erlbaum & INNS Press, Hillsdale, CA, 551-559, 1994.

[30] R.D. Mason & W. Robertson, "Mapping Hierarchical Neural Networks to VLSI Hardware," *Neural Networks*, **8**(6), 905-913, 1995.

[31] G.A. Miller, "The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information", *Psych. Rev.*, **63**(1), 71-97, 1956.

[32] R.C. Minnik, "Linear-Input Logic," *IRE Trans. on Electr. Comp.*, **10**(1), 6-16, 1961.

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

[34] N.P. Red'kin, "Synthesis of Threshold Circuits for Certain Classes of Boolean Functions," *Kibernetika* **5**(1), 6-9, 1970; English translation in *Cybernetics* **6**(5), 540-544, 1973.

[35] V.P. Roychowdhury, A. Orlitsky & K.-Y. Siu, "Lower Bounds on Threshold and Related Circuits via Communication Complexity," *IEEE Trans. Info. Th.*, **40**(2), 467-474, 1994.

[36] C. Shannon, "The Synthesis of Two-Terminal Switching Circuits," *Bell Sys. Tech. J.*, **28**(1), 59-98, 1949.

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

[38] J.D. Ullman, *Computational Aspects of VLSI*, Comp. Sci. Press, Rockville, MA, 1984.

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

[40] M.R. Walker, S. Haghighi, A. Afghan & L.A. Akers, "Training a Limited-Interconnect, Synthetic Neural IC," in D.S. Touretzky (ed.): *Advances in Neural Info. Proc. Sys. 1*, Morgan Kaufmann, San Mateo, CA, 777-784, 1989.

[41] R.C. Williamson, "ε-Entropy and the Complexity of Feedforward Neural Networks," in R.P. Lippmann, J.E. Moody & D.S. Touretzky (eds.): *Advances in Neural Info. Proc. Sys. 3*, Morgan Kaufmann, San Mateo, CA, 946-952, 1990.

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

[43] R.S. Zemel, "Minimum Description Length Analysis," in M.A. Arbib (ed.): *The Handbook of Brain Theory and Neural Networks*, MIT Press, Cambridge, MA, 572-575, 1995.

[44] B.-T. Zhang & H. Mühlenbein, "Genetic Programming of Minimal Neural Networks Using Occam's Razor," *Tech. Rep. GMD 734*, Schloß Birlinghoven, Sankt Augustin, Germany, 1993 (also in *Complex System*, **7**(3), 199-220, 1993).

Report Number (14)	LA-UR--97-4314
	CONF-921235--
Publ. Date (11)	199803[redacted]
Sponsor Code (18)	DOE/HR , XF
UC Category (19)	UC-905 , DOE/ER