CEAI, Vol. 4, No. 1, pp. 33-43, 2002 Printed in Romania

ON OPTIMAL COMPUTATIONS USING PERCEPTRONS

Valeriu BEIU

Washington State University
School of Electrical Engineering & Computer Science
102 Spokane Street (EME), P.O. Box 642752, Pullman, WA 99164-2752, USA
E-mail: vbeiu@eecs.wsu.edu

Abstract: *This paper discusses optimal solutions for implementing arbitrary Boolean functions using perceptrons. It starts by presenting neural structures and their biological inspirations, while mentioning the simplifications leading to artificial neural networks. The state-of-the-art when using neural networks as universal approximators, as well as size optimal perceptron solutions are shortly overviewed. Afterwards we detail a result of Horne and Hush (1994), showing that a neural network of perceptrons restricted to fan-in 2 can implement arbitrary Boolean functions, but requires* $\mathbf{O}(2^n/n)$ *perceptrons in* $\mathbf{O}(n)$ *layers. This result is generalised to arbitrary fan-ins, and used to prove that all the relative minimums of size are obtained for sub-linear ('small') fan-in values. On one side, this result shows a limitation of using perceptrons to implement arbitrary Boolean functions. On the other side, the result is in good agreement with hardware (i.e. VLSI implementations), where the area and the delay are directly related to fan-ins (and to the precision of the synaptic weights). The main conclusion is that discrete VLSI-efficient solutions are connectivity (fan-in) limited even when using perceptrons*

Keywords: *Neural networks, Boolean logic, threshold logic, fan-in, precision.*

1. INTRODUCTION

The model we shall discuss wants to duplicate the activity of the human brain. This is made of living neurons composed of a cell body and many outgrowths. One of these is the axon which may branch into several collaterals. The axon is the 'output' of the neuron. The other outgrowths are the dendrites. The ends of the axons from other neurons are connecting to the dendrites through 'spines'. Active pumps in the nerve cell walls push sodium ions outside, while keeping fewer potassium ions inside. Therefore, their tendency is to keep the cell body at a small negative electric potential (−60mV). The electrical balance varies at the exit point of the axon. If the electrical potential of the cell becomes too positive (+10÷15mV), the potential suddenly jumps to about +60mV. After a short delay (2÷3ms) the potential returns to the normal negative value (−60mV). This change of potentials is sequential, and is called an action potential. The action potential travels down the axon and its branches (with a speed in the range 1÷10m/s). This variation of potential represents

the signal sent by one neuron to its neighbours.

The generation of the signal is achieved by summing the signals coming from the dendrites. The strength of the action potentials traveling along an axon are identical, nevertheless, the effects to the neighbouring cells are different. This is due to the rescaling effect that takes place at the synapse. Although over-simplified, this description of the living nerve cells is a correct representation of the system.

Formally, 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:

$$f(\mathbf{X}) = \sigma\left(\sum_{i=1}^{\Delta} w_i x_i + \theta\right) \qquad (1)$$

the network is a neural network (NN), with $w_i \in \mathbb{R}$ the synaptic weights, $\theta \in \mathbb{R}$ 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, and can be layered. That is why such a network is also known as a multilayer feedforward neural network. The connecting weights are quite important, as it is their modification that allows the NN to 'learn'. The basic idea is to present the examples to the NN and change the weights in such a way as to improve the results (i.e., the outputs of the NN will be 'closer' to the desired values).

The cost functions used to characterise NNs 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).

In the last decade, the tremendous impetus of VLSI technology has made neurocomputer design a lively research topic. Hundreds of designs have been already build, while a few are available as commercial products. Still, we are far from the main objective, as can be clearly seen from Fig. 1, where the horizontal axis represents the number of synapses (i.e., the connectivity), while the vertical axis represents the 'power of computation' in connections per second (CPS). It becomes clear that biological NNs are far ahead of digital, analog and even future optical implementations.

For VLSI implementations 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 have taken into account other cost functions, which can be linked to VLSI implementations by the assumptions one makes on how the area of a chip scales with the weights and the thresholds (Beiu, 1996b, 1996c, 1998).

It is worth emphasizing here that it is highly desirable (if not required) to drastically limit the range of parameter values for VLSI implementations (Wray and Green, 1995), be they digital or analog, because:

- the maximum value of the *fan-in* (Hammerstrom, 1988; Walker *et al.*, 1989), andthe maximal ratio between the largest and the smallest *weight,*
- cannot grow over a certain (technological) limit (Bruck and Goodmann, 1988; Drăghici and Sethi, 1997).

The paper will start by overviewing many results about the approximation capabilities of NNs, and details upper and lower bounds on the size of NNs of perceptrons (i.e., threshold gates). We will show that both Boolean and threshold gate (TG) circuits (TGCs) require exponential size for implementing arbitrary Boolean functions (BFs), while there are optimal TGCs having sub-linear ('small') fan-ins and low precision. Such results are in agreement with silicon implementations (which lack the third dimension of the biological nets) having limited fan-in and reduced precision. Several conclusions are ending the paper.

2. PREVIOUS RESULTS

NNs have been experimentally shown to be quite effective in many applications (see Applications of Neural Networks in (Arbib, 1995), together with Part F: Applications of Neural Computation and Part G: Neural Networks in Practice: Case Studies from (Fiesler and Beale, 1996)). This success has led researchers to undertake a rigorous analysis of their mathematical properties and has generated two directions of research for finding: existence/constructive proofs for the *'universal approximation problem'*;

tight bounds on the *size* of the NNs solving the approximation problem.

Both aspects will be shortly discussed further.

2.1. *Neural Networks as Universal Approximators*

One line of research has concentrated on the approximation capabilities of NNs (Blum and Li, 1991; Ito, 1991). It was started in 1987 by Hecht-Nielsen (1987) and Lippmann (1987) who, together with LeCun (1987), were probably the first to recognise that the specific format from (Sprecher, 1965, 1966) of the form:

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

of Kolmogorov's superpositions $f(x_1, \ldots, x_n) = \sum_{q=1}^{2n+1} \Phi_q (y_q)$ (Kolmogorov, 1957) 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 was given in 1988 by Cybenko (1988, 1989) using a continuous activation function, and was independently presented by Irie and Miyake (1988). Similar results for radial basis functions were shortly reported (Hartman et al., 1989; Poggio and Girosi, 1989). 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 presented later (see also (Scarselli and Tsoi, 1998)):

- Funahashi (1989) proved the same result in a more constructive way, and refined the use of Kolmogorov's theorem in (Hecht-Nielsen, 1987), giving an approximation result for two-hidden-layer NNs;
- Hornik et al. (1989) showed that the continuity requirement for the output function can partly be removed;
- Hornik et al. (1990) also proved that a NN can approximate simultaneously a function and its derivative;
- Park and Sandberg (1991, 1993) used radial basis functions in the hidden layer, and gave an almost constructive proof;
- Hornik (1991) showed that the continuity requirement can be completely removed, the activation function having to be 'bounded and non-constant';
- Geva and Sitte (1992) proved that four-layered NNs with sigmoid activation function are universal approximators;
- Kůrková (1992), and Kůrková et al. (1997), have demonstrated the existence of approximate superposition representations within the constraints of NNs, i.e. ψ and Φq can be approximated by Σar σ(br x + cr), where σ is an arbitrary activation sigmoidal function; depending on approximation, the size of the resulting NNs is between nm (m+1) and m 2(m+1) n;
- Mhaskar and Micchelli (1992, 1994) approach was based on the Fourier series of the function, by truncating the infinite sum to a finite set, and rewriting eikx in terms of the activation function (which now has to be periodic);
- Koiran (1993) presented a new proof on the line of Funahashi's (1989), but more general in that it allows the use of units with 'piecewise continuous' activation functions;
- Leshno et al. (1993) 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 (1993) later proved 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 (1993) showed that the universal approximation theorem also holds for trajectories;
- Sprecher (1993) has demonstrated that there are universal hidden layers that are independent of n;
- Barron (1993) described spaces of functions that can be approximated by the relaxed algorithm of Jones (1992), using functions computed by single-hidden-layer NNs;

- Ito (1994) gave an elementary constructive method improving on the estimates of Kůrková (1992), the size of the resulting NNs being now between nm and m n.

All these results—with the exception of (Barron, 1993; Koiran, 1993; Park and Sandberg, 1991, 1993) - were obtained "provided that sufficiently many hidden units are available" (i.e., with no claims of size minimality). Later, more constructive solutions have been obtained for NN having very small depth (Kůrková, 1992; Ito, 1994; Katsuura and Sprecher, 1994; Nees, 1994, 1996), but their size—or the required precision - grows fast with respect to the number of dimensions n.

Two important results are those of:

- Attali and Pagès (1997), who have given an elementary proof based on the Taylor expansion and the Vandermonde determinant, yielding bounds for the design of the hidden layer, and convergence results for the derivatives;
- Sprecher (1996a, 1996b, 1997), who gave an explicit numerical algorithm for superpositions.

2.2. *Threshold Gate Circuits*

The other line of research was to find the smallest size NN that can realise an arbitrary function given a set of m vectors from $\mathbb{R}^n$. Many results have been obtained for TGs (Myhill and Kautz, 1961). The first lower bound on the size of a TGC for "almost all" n-ary BFs (f: $\mathbb{B}^n \to \mathbb{B}^n$) was given by Neciporuk (1964):

$$size \geq 2 \cdot (2^n / n)^{1/2} \quad (3)$$

Later a very tight upper bound was proven in depth = 4 (Lupanov, 1973):

$$size \leq 2 \cdot (2^n / n)^{1/2} \times \{1 + \Omega[(2^n / n)^{1/2}]\} \quad (4)$$

A similar existence exponential lower bound of $\Omega(2^{n/3})$ for arbitrary BFs can be found in (Siu et al., 1991), which also gives bounds for many particular but important BFs (see also (Roychowdhury et al., 1994)).

For classification problems (f : $\mathbb{R}^n \to \mathbb{B}^k$), the first result was that a NN of depth = 3 and size = m−1 (here m is the number of examples which have to be classified) could compute an arbitrary dichotomy, i.e. a classification into one class (k = 1). The main improvements have been:

- Baum (1988) presented a TGC with one hidden layer having ⌈m/n⌉ neurons capable of realising an arbitrary dichotomy on a set of m points in general position in IRn; if the points are on the corners of the n-dimensional hypercube, m−1 nodes are still needed;
- a slightly tighter bound of only ⌈1+(*m*−2)/*n*⌉ 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 in (Huang and Huang, 1991); the *m*−1 nodes condition was shown to be the least upper bound needed;
- Arai (1993) showed that m−1 hidden neurons are necessary for arbitrary separability, but improved the bound for the dichotomy problem to m/3 (without any condition);
- Beiu (1996a) has detailed the following existence lower and upper bounds: 2mlogm/n 2 < size < 2mlogm/(n 2logn), by estimating the entropy of the data-set;
- Beiu and De Pauw (1997) have presented several improvements on the results of Beiu (1996a), by proving two new bounds 2m/(nlogn) < size < 1.44m/n (see also (Beiu and Drăghici, 1997; Beiu et al., 1998)).

Other existence lower bounds for the arbitrary dichotomy problem (Hassoun, 1995; Paugam-Moisy, 1992) are:

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

One study (Bulsari, 1993) has tried to unify these two lines of research 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, and under piecewise linear

approximations (using ramps instead of sigmoids).

2.3. *Boolean Functions*

The particular case of BFs has been intensively studied (Parberry, 1994). A general solution for synthesising one BF with fan-in = 2 AND-OR gates is based on the classical construction developed by Shannon (1949). It was later extended to the multioutput case, and modified to apply to NN by Horne and Hush (1994):

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

Proof Decompose each output BF into two subfunctions using Shannon's decomposition (Shannon, 1949):

$$f(x_1, x_2, \ldots, x_n) = \bar{x}_1 f_0(x_2, \ldots, x_n) + x_1 f_1(x_2, \ldots, x_n)$$

By doing this recursively, the output BFs will be implemented by binary trees. To eliminate most of the lower level nodes, replace 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:

$$size_{upper} = 3\mu \cdot \sum_{i=0}^{n-q-1} 2^i = 3\mu \cdot (2^{n-q} - 1) \quad (5)$$

$$depth_{upper} = 2(n-q)$$

The subfunctions now depend on q variables, and the lower subnetwork that computes all the possible BFs of q variables has:

$$size_{lower} = 3 \cdot \sum_{i=1}^{q} 2^{2^i} < 4 \cdot 2^{2^q} \quad (6)$$

$$depth_{lower} = 2q$$

(see Fig. 2 from (Horne and Hush, 1994)).

That q which minimises the size of the two subnetworks:

$$size_{BFs} = size_{upper} + size_{lower} \quad (7)$$

is determined by solving $\partial(size_{BFs}) / \partial q = 0$:

$$q \approx \log\{n + \log\mu - 2\log(n + \log\mu)\} \quad (8)$$

By substituting (8) in (5) and (6):

$$size^*_{BFs}(n, \mu) \approx 3\mu \cdot 2^{n-q} = 3\mu \cdot 2^n / (n + \log\mu) = \mathbf{\Theta}\{\mu \cdot 2^n / (n + \log\mu)\} \quad (9)$$

is determined.

The depth is $depth_{upper} + depth_{lower} = 2(n-q) + 2q = 2n = \mathbf{O}(n)$.

3. OPTIMAL PERCEPTRON SOLUTIONS

It is well known that implementing arbitrary BFs using classical Boolean gates (i.e., AND, OR, and NOT gates) requires exponential size circuits (of logarithmic depth). As has been shown in the previous section, the known bounds for size are also exponential if TGCs are used to solve arbitrary BFs in constant depth. These bounds reveal exponential gaps (between the two implementations of arbitrary BFs: using Boolean gates, and respectively TGs). These also suggest that TGCs with more layers might have a smaller size (depth ≠ small const. (Beiu, 1997a, 1997b; Beiu and Makaruk, 1998)).

We start from the classical construction developed by Shannon (1949) for synthesising one BF with fan-in = 2 AND-OR gates, and generalise Proposition 1 to arbitrary fan-in.

> *Proposition 2. Arbitrary Boolean functions of the form* $f: \{0, 1\}^n \to \{0, 1\}^\mu$ *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 and Hush (1994) 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 = Δ OR gates. This first decomposition step eliminates $\Delta-1$ variables and generates a tree of:

$$depth = 1 + \lceil(\Delta - 1)/\log\Delta\rceil^{*/},$$

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

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

Repeating this procedure recursively k times, we have:

$$depth_{upper} = k \cdot \{1 + \lceil (\Delta - 1)/\log\Delta \rceil\}, \qquad (10)$$

$$size_{upper}$$
$$= \{2^{\Delta-1} + \lceil (2^{\Delta-1} - 1)/(\Delta - 1) \rceil\} \cdot \sum_{i=0}^{k-1} 2^{i(\Delta-1)}$$
$$= size \cdot \{2^{k(\Delta-1)} - 1\}/(2^{\Delta-1} - 1)$$
$$\cong 2^{k(\Delta-1)} (1 + 1/\Delta)$$
$$\approx 2^{k\Delta - k}, \qquad (11)$$

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:

$$depth_{lower}$$
$$= \lfloor (n - k\Delta)/\Delta \rfloor \cdot \{1 + \lceil (\Delta - 1)/\log\Delta \rceil\}, \qquad (12)$$

$$size_{lower}$$
$$= \{2^{\Delta-1} + \lceil (2^{\Delta-1} - 1)/(\Delta - 1) \rceil\} \times$$
$$\times \sum_{i=1}^{\lfloor n/\Delta \rfloor - k} 2^{2^{n - k\Delta - i\Delta}}$$
$$= size \cdot \{2^{2^0} + 2^{2^\Delta} + \ldots + 2^{2^{n-(k+1)\Delta}}\}$$
$$< (size + 1) \cdot 2^{2^{n-(k+1)\Delta}} \qquad (13)$$
$$\approx 2^\Delta \cdot 2^{2^{n - k\Delta - \Delta}}. \qquad (14)$$

The inequality (13) 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 shall prove it for $\alpha+1$:

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

(due to hypothesis), thus:

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

and computing the logarithm of the left side:

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

From (10) and (12) we can estimate $depth_{BFs}$ as:

$$depth_{BFs}$$
$$= \{k + (n - k\Delta)/\Delta\} \cdot \{1 + \lceil (\Delta - 1)/\log\Delta \rceil\}$$
$$= (n/\Delta) \cdot (\Delta/\log\Delta + 1)$$
$$= n/\log\Delta$$
$$= \mathbf{O}(n/\log\Delta), \qquad (15)$$

and from (11) and (13) we estimate $size_{BFs}$ as:

$$size_{BFs} = \mu \cdot size \cdot [2^{k(\Delta-1)} - 1]/(\Delta - 1)$$
$$+ (size + 1) \cdot 2^{2^{n-(k+1)\Delta}}$$
$$\approx \mu \cdot 2^{k\Delta - k} + 2^\Delta \cdot 2^{2^{n - k\Delta - \Delta}} \qquad (16)$$

concluding the proof.

Proposition3 Arbitrary Boolean functions of the form $f: \{0, 1\}^n \to \{0, 1\}^\mu$ can be implemented by neural networks of perceptrons, where all the critical points of $size_{BFs}(\mu, n, k, \Delta)$, are relative minimum 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 of $size_{BFs}$ given by (16) we compute $\partial size_{BF}/\partial k = 0$:

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

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

$$\log\beta + 2\gamma - k - n = 2^{n - \gamma - \Delta}, \qquad (17)$$

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

We can verify this result (obtained by approximating the partial derivative) by computing with finite differences:

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

$$size \cdot \{\mu \cdot 2^{k\Delta - k} - 2^{2^{n-k\Delta-\Delta}}\} = 0$$

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

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

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

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

$$\approx n - \Delta - \log(\gamma + \log\mu), \qquad (18)$$

which has the same approximate solution:

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

Starting again from (16), we compute $\partial size_{BFs}/\partial\Delta = 0$:

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

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

$$\mu k \cdot 2^{\gamma - k}$$

$$= k(\ln 2)\cdot 2^{n-\gamma}\cdot 2^{2^{n-\gamma-\Delta}} - 2^{\Delta}\cdot 2^{2^{n-\gamma-\Delta}}$$

$$\mu k \cdot 2^{\gamma-k}\cdot 2^{\gamma-n}$$

$$= k(\ln 2)\cdot 2^{2^{n-\gamma-\Delta}} - 2^{\Delta}\cdot 2^{\gamma-n}\cdot 2^{2^{n-\gamma-\Delta}}$$

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

$$(\mu/\ln 2)\cdot 2^{2\gamma-k-n}$$

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

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

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

i.e., an equation similar to (17).

All of these show quite clearly that the critical points are situated somewhere in the (close) vicinity of the parabola $k\Delta \approx n - \log(n + \log\mu)$. It follows that size-optimal TGCs can be obtained for sub-linear ('small') fan-ins, i.e. fan-ins ranging from small constants to at most n – log 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. The results of those simulations can be seen in Fig. 1. From Fig. 1(a), (b), and (c), it seems that k and Δ have roughly the same influence on $size_{BFs}$. The parabola-like discrete curves are approximations of $k\Delta \approx n - \log(n + \log\mu)$, and can be seen in Fig. 1(d), (e), and (f).

Remark It is to be mentioned that the other relative minima (on, or in the vicinity of the parabola $k\Delta = n - \log n$) might be of more practical interest, as leading to shallower networks, i.e., having fewer layers: $n / \log\Delta$ (instead of n).

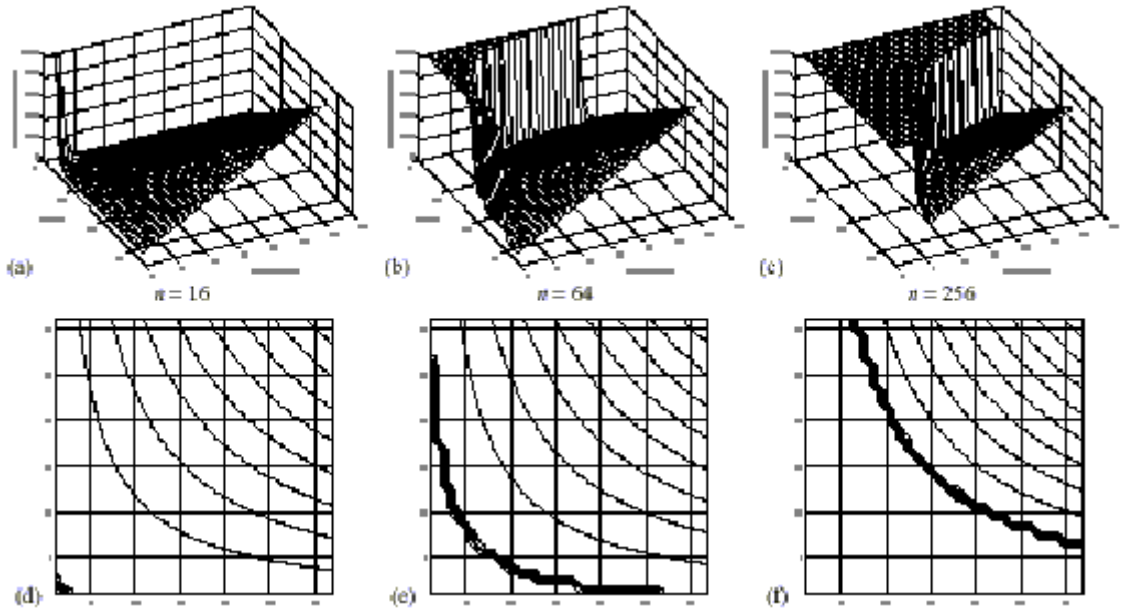

Fig. 1. The size (log scale) of neural networks implementing arbitrary Boolean functions. *Size* versus *fan in* and the *k* parameter: (a) n=16; (b) n=64; (c) n=256 (clipped at 2^{1000}). The contour plots for the same cases: (d) n=16; (e) n=64; (f) n=256.

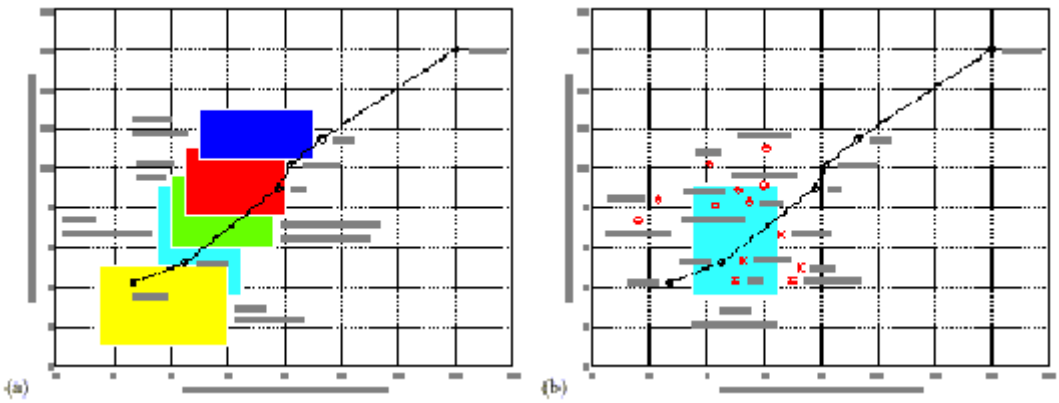

Fig. 2. Biological neural networks and different hardware alternatives for implementing artificial neural networks.
(a) An updated and enhanced version inspired from Glesner and Pochmuller (1994).
(b) Detail showing digital neurochips as circles and classical computers as crosses (for detail see (Beiu, 1996c)).

4. CONCLUSIONS

The main conclusion of this paper is that size optimal solutions for implementing arbitrary Boolean functions using TGs (perceptrons) are obtained for TGs having sub-linear fan-ins. In general, arbitrary BFs can be implemented using either classical Boolean gates, or TGs (perceptrons). In both cases, the size is exponential, but there is an exponential gap in between the optimal sizes given by these two implementations. All of these show that in the space of all possible solutions, there are very interesting fan-in dependent depth-size (or area-delay for VLSI implementations) tradeoffs, as well as size optimal TGCs having relatively 'small' fan-in values. Still, finding these size optimal solutions is not at all obvious, and requires a lot of effort.

These results also suggest that removing the analog beaviour of the neurons by substituting the sigmoid (non-linear) activation function σ with a hard limiter reduces significantly their computation abilities. When compared to biological neural networks, perceptron based hardware implementations (being connectivity and precision limited) will not able to compensate by their higher computing speeds (see Fig. 2(a)). We claim that the brain does not optimise energy and power - like engineers do when designing integrated circuits—and probably trades the slower individual speeds (thus, reducing power!) of its elementary analog computing elements, for their higher connectivity (larger fan-ins).

5. REFERENCES

[1] Arai, M. - "Bounds on the number of hidden units in binary-valued three-layer neural networks", *Neural Networks*, vol. 6, pp. **855-860**, 1993.

[2] Arbib, M.A. - "The Handbook of Brain Theory and Neural Networks", Cambridge, MA: MIT Press, 1995.

[3] Attali J.-G. and Pagès, G. - "Approximations of functions by a multilayer perceptron: A new approach", *Neural Networks*, vol. 10, pp. **1069-1081**, 1997.

[4] Barron, A.R. - "Universal approximation bounds for superpositions of a sigmoidal function", *IEEE Trans. Info. Theory*, vol. 39, pp. **930-945**, 1993.

[5] Baum, E.B. - "On the capabilities of multilayer perceptrons", J. Complexity, vol. 4, pp. **193-215**, 1988.

[6] Beiu, V. - "Entropy bounds for classification algorithms", *Neural Network World*, vol. 6, pp. **497-505**, 1996a.

[7] Beiu, V. - "Optimal VLSI implementation of neural networks, in Taylor", J.G. (ed.), *Neural Networks and Their Applications*, Chichester, UK: John Wiley, Chap. 18, pp. **255-276**, 1996b.

[8] Beiu, V. - "Digital integrated circuit implementations", in Fiesler, E., and Beale, R. (eds.), *Handbook of Neural Computation*, New York, NY: Inst. of Physics, Chap. E1.4, 1996c.

[9] Beiu, V. - "Constant fan-in digital neural networks are VLSI-optimal", in Ellacott, S.W., Mason, J.C., and Anderson, I.J. (eds.), *Mathematics of Neural Networks: Models, Algorithms and Applications*, Boston, MA: Kluwer Academic, Chap. 12, pp. **89-94**, 1997a.

[10] Beiu, V. - "When constants are important", in Dumitrache, I. (ed.), Proc. *Intl. Conf. on Control System and Computer Science CSCS-11*, Bucharest, Romania: UPB Press, vol. 2, pp. **106-111**, 1997b.

[11] Beiu, V. - "On the circuit and VLSI complexity of threshold gate COMPARISON", *Neurocomputing*, vol. 19, pp. **77-98**, 1998.

[12] Beiu, V. and De Pauw, T. - "Tight bounds on the size of neural networks for classification problems", in Mira, J., Moreno-Díaz, R., and Cabestany, J. (eds.), *Biological and Artificial Computation*, Berlin, Germany: Springer-Verlag, pp. **743-752**, 1997.

[13] Beiu, V. and Drăghici, S. - "Limited weights neural networks: very tight entropy based bounds", in Pearson, D.W. (ed.), *Proc. Intl. ICSC Symp. on Soft Computing SOCO'97*, Millet, Canada: ICSC Acad. Press, pp. **111-118**, 1997.

[14] Beiu, V. and Makaruk, H.E. - "Deeper sparser nets can be optimal", *Neural Processing Letters*, vol. 8, pp. **201-210**, 1998.

[15] Beiu, V., Drăghici, S. and & De Pauw, T. - "A constructive approach to calculating lower entropy bounds", *Neural Processing Letters*, vol. 9, pp. **1-12**, 1998.

[16] Blum, E. and Li, K. - "Approximation theory and feedforward networks", *Neural Networks*, vol. 4, pp. **511-515**, 1991.

[17] Bruck, J. and Goodmann, J.W. - "On the power of neural networks for solving hard problems", in Anderson, D.Z. (ed.), *Neural Information Processing Systems*, New York, NY: AIPress, pp. **137-143**, 1988 (also in J. Complexity, vol. 6, pp. **129-135**, 1990).

[18] Bulsari, A. - "Some analytical solutions to the general approximation problem for feedforward neural networks", *Neural Networks*, vol. 6, pp. **991-996**, 1993.

[19] Cybenko, G. - "Continuous valued neural networks with two hidden layers are sufficient", Tech. Rep., Maths. Dept., Tufts Univ., Medford, 1988.

[20] Cybenko, G. - "Approximations by superpositions of a sigmoid function", *Math. of Control, Signals and Systems*, vol. 2, pp. **303-314**, 1989.

[21] Drăghici, S. and Sethi, I.K. - "On the possibilities of the limited precision weights neural networks in classification problems", in Mira, J., Moreno-Díaz, R., and Cabestany, J. (eds.), *Biological and Artificial Computation*, Berlin, Germany: Springer-Verlag, pp. **753-762**, 1997.

[22] Fiesler, E. and Beale, R. - "Handbook of Neural Computation", New York, NY: Inst. of Physics, 1996.

[23] Funahashi, K.-I. - "On the approximate realization of continuous mapping by neural networks", *Neural Networks*, 2, pp. **183-192**, 1989.

[24] Funahashi, K.-I. and Nakamura, Y. - "Approximation of dynamical systems by continuous time recurrent neural networks", *Neural Networks*, 6, pp. **801-806**, 1993.

[25] Geva, S. and Sitte, J. - "A constructive method for multivariate function approximation by multilayered perceptrons", *IEEE Trans. Neural Networks*, 3, pp. **621-623**, 1992.

[26] Glesner, M. and Pöchmüller, W. - "Neuro-computers – An Overview of Neural Networks in VLSI", London, UK: Chapman and Hall, 1994.

[27] Hammerstrom, D. - "The connectivity analysis of simple association -or- how many connections do you need", in Anderson, D.Z. (ed.), *Neural Information Processing Systems*, New York, NY: AIPress, pp. **338-347**, 1988.

[28] Hartman, E., Keeler, J.D. and Kowalski, J.M. - "Layered neural networks with gaussian hidden units as universal approximations", *Neural Computation*, vol. 2, pp. **210-215**, 1989.

[29] Hassoun, M.H. - "Fundamentals of Artificial Neural Networks", Cambridge, MA: MIT Press, 1995.

[30] Hecht-Nielsen, R. - "Kolmogorov's mapping neural network existence

theorem", in *Proc. IEEE Intl. Conf. on Neural Networks ICNN'87*, New York, NY: IEEE CS Press, vol. 3, pp. **11-14**, 1987.

[31] Horne, B.G. and Hush, D.R. - "On the node complexity of neural networks", *Neural Networks*, vol. 7, pp. **1413-1426**, 1994.

[32] Hornik, K. - "Approximation capabilities of multilayer feedforward networks", *Neural Networks*, vol. 4, pp. **251-257**, 1991.

[33] Hornik, K. - "Some new results on neural network approximation", *Neural Networks*, vol. 6, pp. **1069-1072**, 1993.

[34] Hornik, K., Stinchcombe, M. and White, H. - "Multilayer feedforward neural networks are universal approximators", *Neural Networks*, vol. 2, pp. **359-366**, 1989.

[35] Hornik, K., Stinchcombe, M. and White, W. - "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks", *Neural Networks*, vol. 3, pp. **551-560**, 1990.

[36] Huang, S.-C. and Huang, Y.-F. - "Bounds on the number of hidden neurons of multilayer perceptrons in classification and recognition", *IEEE Trans. Neural Networks*, vol. 2, pp. **47-55**, 1991.

[37] Irie, B. and Miyake, S. - "Capabilities of three-layered perceptrons", in *Proc. IEEE Intl. Conf. on Neural Networks ICNN'88*, New York, NY: IEEE CS Press, vol. 1, pp. **641-648**, 1988.

[38] Ito, Y. - "Approximation of functions on a compact set by finite sums of sigmoid functions without scaling", *Neural Networks*, vol. 4, pp. **817-826**, 1991.

[39] Ito, Y., - "Approximation capabilities of layered neural networks with sigmoid units on two layers", *Neural Computation*, vol. 6, pp. **1233-1243**, 1994.

[40] Jones, L.K. - "A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training", Ann. Stat., vol. 20, pp. **608-613**, 1992.

[41] Katsuura, H. and Sprecher, D.A. - "Computational aspects of Kolmogorov's superposition theorem", *Neural Networks*, vol. 7, pp. **455-461**, 1994.

[42] Koiran, P. - "On the complexity of approximating mappings using feedforward networks", *Neural Networks*, vol. 6, pp. **649-653**, 1993.

[43] Kolmogorov, A.N. - "On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition", Dokl. Akad. Nauk SSSR, vol. 114, pp. **953-956**, 1957 (English transl., Trans. American Math. Soc., vol. 2, pp. **55-59**, 1963).

[44] Kůrková, V. - "Kolmogorov's theorem and multilayer neural networks", *Neural Networks*, vol. 5, pp. **501-506**, 1992.

[45] Kůrková, V., Kainen, P.C. and Kreinovich, V. "Estimates of the number of hidden units and variations with respect to half-spaces", *Neural Networks*, vol. 10, pp. **1061-1068**, 1997.

[46] LeCun, Y. - "Models connexionistes de l'apprentisage", *M.Sc. dissertation*, Univ. Pierre et Marie Curie, Paris, France, 1987.

[47] Leshno, M., Lin, V.Y., Pinkus, A. and Schocken, S. - "Multilayer feedforward neural networks with a nonpolynomial activation function can approximate any function", *Neural Networks*, vol. 6, pp. **861-867**, 1993.

[48] Lippmann, R.P. - "An introduction to computing with neural nets", *IEEE ASSP Mag.*, vol. 4, pp. **4-22**, 1987.

[49] Lupanov, O.B. - "The synthesis of circuits from threshold elements", *Problemy Kibernetiki*, vol. 20, pp. **109-140**, 1973.

[50] Mhaskar, H.N. and Micchelli, C. - "Approximation by superposition of sigmoidal and radial basis functions", *Advs. Appl. Maths.*, vol. 13, pp. **350-373**, 1992.

[51] Mhaskar, H.N. and Micchelli, C. - "Dimension independent bounds on the degree of approximation by neural networks", *IBM J. Res. and Dev.*, vol. 38, pp. **277-283**, 1994.

[52] Myhill, J. and Kautz, W.H. - "On the size of weights required for linear-input switching functions", *IRE Trans. Electr. Comp.*, vol. 10, pp. **288-290**, 1961.

[53] Neciporuk, E.I. - "The synthesis of networks from threshold elements", *Soviet Mathematics - Doklady*, vol. 5, pp. **163-166**, 1964 (English transl., *Automation Express*, vol. 7, pp. 27-32, and vol. 7, pp. **35-39**, 1964).

[54] Nees, M. - "Approximate versions of Kolmogorov's superposition theorem, proved constructively", J. Comp. & Appl. Math., vol. 54, pp. **239-250**, 1994.

[55] Nees, M. - "Chebyshev approximation by discrete superposition: Application to neural networks", *Advs. Comp. Maths*., vol. 5, pp. **137-152**, 1996.

[56] Parberry, I. - "Circuit Complexity and Neural Networks", Cambridge: MIT Press, 1994.

[57] Park, J. and Sandberg, I.W. - "Universal approximation using radial-basis-function networks", *Neural Computation*, vol. 3, pp. **246-257**, 1991.

[58] Park, J. and Sandberg, I.W. - "Approximation and radial-basis-function networks", *Neural Computation*, vol. 5, pp. **305-316**, 1993.

[59] Paugam-Moisy, H. - "Optimisation des réseaux des neurones artificiels", *PhD dissertation*, LIP, École Normale Supérieure de Lyon, Lyon, France, 1992.

[60] Poggio, T. and Girosi, F. - "A theory of networks for approximation and learning", *Tech. Rep. AI Memo 1140*, MIT, 1989 (short version in Proc. IEEE, 78, pp. **1481-1497**, 1990).

[61] Roychowdhury, V.P., Orlitsky, A. and Siu, K.-Y. - "Lower bounds on threshold and related circuits via communication complexity", *IEEE Trans. Info. Theory*, vol. 40, pp. **467-474**, 1994.

[62] Scarselli, F. and Tsoi, A.C. - "Universal approximation using feedforward neural networks: a survey of some existing methods, and some new results", *Neural Networks*, vol. 11, pp. **15-37**, 1998.

[63] Shannon, C. - "The synthesis of two-terminal switching circuits", *Bell Sys. Tech. J.*, vol. 28, pp. **59-98**, 1949.

[64] Siu, K.-Y., Roychowdhury, V.P. and Kailath, T. - "Depth-size tradeoffs for neural computations", *IEEE Trans. Comp.*, vol. 40, pp. **1402-1412**, 1991.

[65] Sprecher, D.A. - "On the structure of continuous functions of several variables", *Trans. American Math. Soc.*, vol. 115, pp. **340-355**, 1965.

[66] Sprecher, D.A. - "On the structure of representations of continuous functions of several variables as finite sums of continuous functions of one variable", "Proc. American Math. Soc.", vol. 17, pp. **98-105**, 1966.

[67] Sprecher, D.A. - "A universal mapping for Kolmogorov's superposition theorem", *Neural Networks*, vol. 6, pp. **1089-1094**, 1993.

[68] Sprecher, D.A. - "A numerical implementation of Kolmogorov's superpositions", *Neural Networks*, vol. 9, pp. 765-772, 1996a.

[69] Sprecher, D.A. - "A numerical construction of a universal function for Kolmogorov's superpositions", *Neural Network World*, vol. 6, pp. **711-718**, 1996b.

[70] Sprecher, D.A. - "A numerical implementation of Kolmogorov's superpositions II", *Neural Networks*, vol. 10, pp. **447-457**, 1997.

[71] Walker, M.R., Haghighi, S., Afghan, A. and Akers, L.A. - "Training a limited-interconnect, synthetic neural IC", in Touretzky, D.S. (ed.), Advances in Neural *Information Processing Systems*, San Mateo, CA: Morgan Kaufmann, pp. **777-784**, 1989.

[72] Williamson, R.C. - "ε-entropy and the complexity of feedforward neural networks", in Lippmann, R.P., Moody, J.E., and Touretzky D.S. (eds.), *Advances in Neural Information Processing Systems*, San Mateo, CA: Morgan Kaufmann, pp. **946-952**, 1990.

[73] Wray, J. and Green, G.G.R. - "Neural net works, approximation theory, and finite precision computation", *Neural Networks*, vol. 8, pp. **31-37**, 1995.