LA-UR- 98-2504

Approved for public release; distribution is unlimited.

Title: 2D NEURAL HARDWARE VERSUS 3D BIOLOGICAL ONES

CONF-980966--

Author(s): Valeriu Beiu, NIS-1

Submitted to: International Symposium on Neural Computation, Vienna, Austria

Los Alamos
NATIONAL LABORATORY

DISCLAIMER

DISCLAIMER

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

2D Neural Hardware Versus 3D Biological Ones

Valeriu Beiu [1]

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

Abstract*—*This paper will presents important limitations of hardware neural nets as opposed to biological neural nets (i.e. the real ones). We will start by discussing neural structures and their biological inspirations, while mentioning the simplifications leading to artificial neural nets. Going further, the focus will be on hardware constraints. We will present recent results for three different alternatives of implementing neural networks: digital, threshold gate, and analog, while the area and the delay will be related to neurons' fan-in and weights' precision. Based on all of these, it will be shown why hardware implementations cannot cope with their biological inspiration with respect to their 'power of computation': the mapping onto silicon lacking the third dimension of biological nets. This translates into reduced fan-in, and leads to reduced precision. The main conclusion is that we are faced with the following alternatives: (i) try to cope with the limitations imposed by silicon, by speeding up the computation of the elementary 'silicon' neurons; (ii) investigate solutions which would allow us to use the third dimension, e.g. using optical interconnections.

Keywords*—*neural networks, Boolean functions/circuits, threshold gate circuits, analog circuits, circuit complexity, VLSI complexity, fan-in, size, precision (accuracy).

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 end 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 of 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 travelling along an axon are identical, nevertheless, the effects to the neighbouring cells are different. This is due to the rescaling effect which 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:

$$f(\boldsymbol{x}) = f(x_1, \ldots, x_\Delta) = \sigma\left(\sum_{i=1}^{\Delta} w_i x_i + \theta\right), \qquad (1)$$

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 connection *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 a NN 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 really lively research topic. Hundreds of designs have been already build, and some are available as commercial products. Still, we are far from the main objective as can be clearly seen from *Figure 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. This paper will try to explain why this is the case.

For hardware 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

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

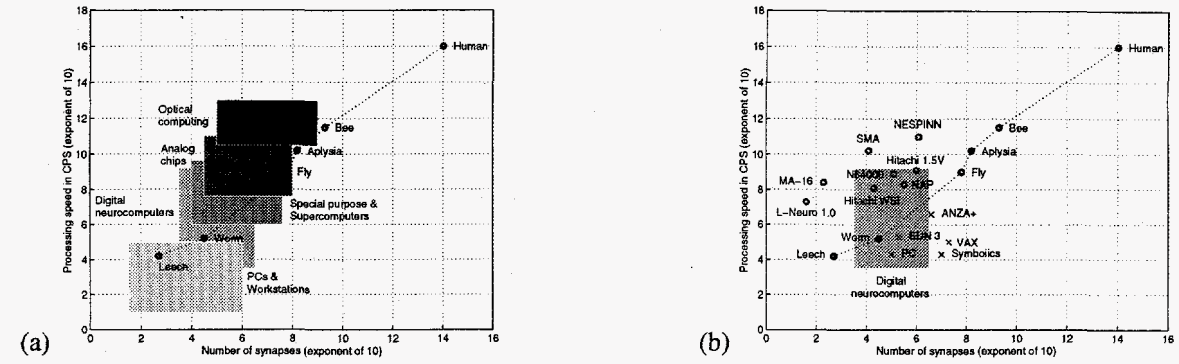

Figure 1. Different hardware alternatives for implementing artificial neural networks: (a) an enhanced version from (Glesner 1994); (b) neurochips (circles) and classical computers (crosses) (for details see (Beiu, 1996b)).

(Abu-Mostafa, 1988; Hammerstrom, 1988; Phatak & Koren, 1994) have taken into account the total *number-of-connections*, or the total *number-of-bits* needed to represent the *weights* and the *thresholds* (Bruck & Goodmann, 1988), or the sum of all the *weights* and the *thresholds* (Beiu *et al.*, 1994). The sum of all the *weights* and the *thresholds*—also applied for defining the minimum-integer *threshold gate* (TG) realisation of a *Boolean function* (BF)—has been recently used under the name of *"total weight magnitude"* in the context of computational learning theory for improving on several standard VC-theory bounds (Bartlett, 1996). A similar definition: $\sum w_i^2$ has also been advocated (Zhang & Mühlenbein, 1993). Such approximations can easily be related to assumptions on how the *area* of a chip scales with the *weights* (Beiu, 1996b, 1997b, 1998b):

- for digital implementation, the *area* scales with the cumulative storage of *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 anyhow desirable to limit the range of parameter values (Wray & Green, 1995) because: (*i*) the maximum value of the *fan-in* (Hammerstrom, 1988; Walker *et al.*, 1989); and (*ii*) the maximal ratio between the largest and the smallest *weight* cannot grow over a certain (technological) limit.

Concerning the *delay*, two well-known models are:

- a simple capacitive one, which assumes that the *delay* is proportional to the input capacitance (*i.e.*, the *delay* proportional to the *fan-in*;
- the exact one assumes a distributed capacitance along any wire, hence the *delay* for propagating a signal is proportional to the *length* of the connecting wire.

The paper starts by overviewing several results dealing with the approximation capabilities of NNs, and details upper and lower bounds on the *size* of *threshold gate circuits* (TGCs). These are followed by solutions which are optimal with respect to some cost function. We show that both Boolean and TGCs require exponential *size* for implementing arbitrary BFs, while there are solutions which can be obtained using low precision and small *fan-ins*. Further, we argue that *size*-optimal solutions of discrete NNs can be obtained only in analog circuitry, but require very high precision and large *fan-ins* (based on a fresh constructive solution for Kolmogorov's superpositions). It follows that the mapping onto silicon—lacking the third dimension of biological nets—translates into 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 & 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: (*i*) existence / constructive proofs for the '*universal approximation problem*'; (*ii*) tight bounds on the *size* needed by the approximation problem.

2.1. Neural Networks as Universal Approximators

One line of research has concentrated on the approximation capabilities of NNs (Blum & Li, 1991; Ito, 1991; Funahashi & Nakamura, 1993; Ito, 1994). 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 in (Sprecher, 1965, 1966) of the form:

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

of Kolmogorov's superpositions (Kolmogorov, 1957) $f(x_1 \dots x_n) = \sum_{q=1}^{2n+1} \Phi_q (y_q)$, 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 & 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 later presented (for details see (Scarselli & Tsoi, 1998), and *Chapter 1* in (Beiu, 1998c)):

- Funahashi (1989) proved the same result but in a more constructive way and also 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 nonconstant';
- Geva and Sitte (1992) proved that four-layered NNs with sigmoid activation function are universal approximators;
- Kůrková (1992) and Kůrková *et al.* (1997) has demonstrated the existence of approximate superposition representations within the constraints of NNs, *i.e.* ψ and Φ_q can be approximated with functions of the form $\sum a_r \sigma(b_r x + c_r)$, where σ is an arbitrary activation sigmoidal function;
- Mhaskar and Micchelli (1992, 1994) approach was based on the Fourier series of the function, by truncating the infinite sum to a finite set, and rewriting e^{ikx} in terms of the activation function (which now has to be periodic);
- Koiran (1993) presented a new proof on the line of Funahashi's proof (Funahashi, 1989), but more general in that it allows the use of units with 'piecewise continuous' activation functions; these include the particular but important case of TGs;
- 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 reported in the literature;
- Hornik (1993) added to these results by proving that: (*i*) if the activation function is locally Riemann integrable and nonpolynomial, the *weights* and the *thresholds* can be constrained to arbitrarily small sets; and (*ii*) if the activation function is locally analytic, a single universal *threshold* will do;
- Funahashi and Nakamura (1993) showed that the universal approximation theorem also holds for trajectories of patterns;
- Sprecher (1993) has demonstrated that there are universal hidden layers that are independent of the number of input variables 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 networks of perceptrons.

All these results—with the partial exception of (Park & Sandberg, 1991; Kůrková, 1992; Barron, 1993; Koiran, 1993; Park & Sandberg, 1993)—were obtained *"provided that sufficiently many hidden units are available"* (*i.e.*, with no claims on the *size* minimality). More constructive solutions have been obtained in very small *depth* later (Katsuura & Sprecher, 1994; Nees, 1994, 1996), but their *size* or the required precision grows fast with respect to the number of dimensions. Recently, Attali and Pagès (1997) 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. An explicit numerical algorithm for superpositions has also been detailed (Sprecher, 1996a, 1996b, 1997).

2.2. Threshold Gate Circuits

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

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

was given by Neciporuk (1964). Later a very tight upper bound was proven in *depth* = 4 (Lupanov, 1973):

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

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 ($f: \mathbb{R}^n \rightarrow \mathbb{B}^k$), the first result was that a NN of *depth* = 3 and *size* = $m - 1$ could compute an arbitrary dichotomy. The main improvements have been:

- Baum (1988) presented a TGC with one hidden layer having $\lceil m/n \rceil$ neurons capable of realising an arbitrary dichotomy on a set of m points *in general position* in $\mathbb{R}^n$; if the points are on the corners of the n-dimensional hypercube, $m - 1$ nodes are still needed;
- a slightly tighter bound of only $\lceil 1 + (m-2)/n \rceil$ neurons in the hidden layer for realising an arbitrary dichotomy on a set of m points which satisfy a more relaxed topological assumption was proven in (Huang & 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 on the inputs);

- Beiu (1996a) has detailed existence lower and upper bounds: $2m\log m / n^2 < size < 2m\log m / n^2 \log n$ (by estimating the entropy of the data-set);
- Beiu and De Pauw (1997) have presented several improvements $2m / (n\log n) < size < 1.44m / n$ (see also (Beiu & Drăghici, 1997; Beiu *et al.*, 1998)).

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

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

One study (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; Beiu, 1998c). Many results have been obtained for particular BFs (Siu *et al.*, 1991; Roychowdhury *et al.*, 1994). A *size*-optimal result for BFs that have exactly m groups of ones in their truth table $IF_{n,m}$ is:

Proposition 1 (Red'kin, 1970) *The complexity realisation (i.e., number of threshold elements) of $IF_{n,m}$ (the class of BFs $f(x_1 x_2 \ldots x_{n-1} x_n)$ 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.

This result—as are all the previous ones—is valid for unlimited *fan-in* TGs. A solution for limited *fan-in* TGCs is:

Proposition 2 (Horne & Hush, 1994) *Arbitrary BFs of the form $f: \{0, 1\}^n \rightarrow \{0, 1\}^\mu$ can be implemented in a NN of perceptrons restricted to fan-in 2 with a node complexity of $\Theta \{\mu 2^n / (n + \log\mu)\}$ and requiring $\boldsymbol{O}(n)$ layers.*

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

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

By doing this recursively, 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:

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

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

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

nodes, and $depth_{lower} = 2q$. That q which minimises the *size* (*i.e.*, $size_{upper} + size_{lower}$) is determined by solving the equation $d (size_{BFs}) / dq = 0$, and gives:

$$q \approx \log\{n + \log\mu - 2\log(n + \log\mu)\}. \quad (7)$$

By substituting eq. 7 in eq. 5 and eq. 6, the minimum *size*:

$$size_{BFs} \approx 3\mu 2^{n-q} = 3\mu \cdot 2^n / (n + \log\mu) \quad (8)$$

is determined. ❑

3. "Optimal" Solutions

It is known that implementing arbitrary BFs using classical Boolean gates (*i.e.*, AND and OR gates) requires exponential *size* circuits. The known bounds for *size* are also exponential if TGCs are used to solve arbitrary BFs (Beiu, 1996b). These bounds reveal exponential gaps, and also suggest that TGCs with more layers (*depth* ≠ small const. (Beiu, 1997a, 1997b; Beiu & Makaruk, 1998)) might have a smaller *size*.

Proposition 3 (Beiu & Makaruk, 1998) *Arbitrary BFs $f: \{0, 1\}^n \rightarrow \{0, 1\}^\mu$ can be implemented in a NN of perceptrons restricted to fan-in Δ in $\boldsymbol{O}(n / \log\Delta)$ layers.*

Sketch of proof We use the approach of Horne & Hush (1994) for the case when the *fan-in* is limited to Δ. Each BF is decomposed in $2^{\Delta-1}$ subfunctions. The $2^{\Delta-1}$ inputs OR gate is decomposed in a Δ-ary tree. This eliminates $\Delta - 1$ variables and generates a *depth* $= 1 + \lceil (\Delta - 1) / \log\Delta \rceil$ tree of *size* $= 2^{\Delta-1} + \lceil (2^{\Delta-1} - 1) / (\Delta - 1) \rceil$. Repeating this procedure recursively k times, we have:

$$depth_{upper} = k \cdot \{1 + \lceil (\Delta - 1) / \log\Delta \rceil\} \quad (9)$$

$$size_{upper} \approx 2^{k\Delta - k}. \quad (10)$$

We now generate all the possible subfunctions of q variables with a subnetwork having:

$$depth_{lower} = \lfloor (n - k\Delta) / \Delta \rfloor \{1 + \lceil (\Delta - 1) / \log\Delta \rceil\} \quad (11)$$

$$size_{lower} < (size + 1) \cdot 2^{2^{n-(k+1)\Delta}} \approx 2^\Delta \cdot 2^{2^{n-k\Delta-\Delta}} \quad (12)$$

From eq. 9 and eq. 11 we can estimate $depth_{BFs}$, and from eq. 10 and eq. 12 $size_{BFs}$ as:

$$depth_{BFs} = (n / \Delta) \cdot (\Delta / \log\Delta + 1) = \boldsymbol{O}(n / \log\Delta) \quad (13)$$

$$size_{BFs} \approx \mu \cdot 2^{k\Delta - k} + 2^\Delta \cdot 2^{2^{n-k\Delta-\Delta}} \quad (14)$$

concluding the proof. ❑

Proposition 4 (Beiu & Makaruk 1998) *All the critical points of the size $size_{BFs}(\mu, n, k, \Delta)$ are relative minimum and are situated in the (close) vicinity of the parabola $k\Delta \approx n - \log(n + \log\mu)$.*

Sketch of proof Equate the partial derivative to zero.

$\partial size_{BFs} / \partial k = 0$, and using the following notations $k\Delta = \gamma$, $\beta = \mu\,(\Delta - 1) / (\Delta \ln 2)$, and taking logarithms of both sides:

$$\log\beta + 2\gamma - k - n \quad = 2^{n-\gamma-\Delta}, \tag{15}$$

with an approximate solution $\gamma \approx n - \log(n + \log\mu)$. Equating $\partial size_{BFs} / \partial\Delta = 0$, leads to the same solution. This shows that the critical points are situated in the vicinity of the parabola $k\Delta \approx n - \log(n + \log\mu)$. ❑

Proposition 5 (Beiu & Makaruk 1998) *The minimum size is obtained for fan-in* $\Delta = 2$.

Sketch of proof Compute $size_{BFs}(n, \mu, k, \Delta)$ for the critical points $k \approx (n - \log n) / \Delta$, and then show that:

$$size^{*}_{BFs}(n, \mu, \Delta + 1) - size^{*}_{BFs}(n, \mu, \Delta) > 0. \tag{16}$$

Hence, the function is monotonically increasing and the minimum is obtained for the smallest *fan-in* $\Delta = 2$. ❑

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

A similar result can be obtained for $IF_{n,m}$, as the first layer is represented by COMPARISONs (*i.e.*, $IF_{n,1}$) which can be decomposed to satisfy the limited *fan-in* condition (Beiu, 1997a, 1997b, 1998a, 1998b; Beiu & Taylor, 1996).

Proposition 6 (Beiu et al., 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* $3 \le \Delta \le n$.

The *size* of the NN implementing one $IF_{n,m}$ function is:

$$size_{IF} \;=\; 2nm \cdot \left\{ \frac{1}{\Delta/2} + \ldots + \frac{1}{(\Delta/2)^{depth_{IF}}} \right\}, \tag{17}$$

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

$$\frac{2n}{\Delta} 2^{\Delta}, \frac{2n/\Delta}{\Delta/2} 2^{\Delta(\Delta/2)}, \ldots, \frac{2n/\Delta}{(\Delta/2)^{depth_{IF}-1}} 2^{\Delta(\Delta/2)^{depth_{IF}-1}} \tag{18}$$

For large m (needed for achieving a certain precision (Beiu, 1998a; Wray & Green, 1995)), and / or large n, the first terms of the sum from eq. 17 will be larger than the equivalent ones from eq. 18. This is equivalent to the trick from (Horne & Hush, 1994), as the lower levels will compute *all the possible functions* using only limited *fan-in* COMPARISONs. Hence, the optimum *size* becomes:

$$size^{*}_{IF} = 2n \cdot \left\{ \sum_{i=1}^{k} \frac{2^{\Delta(\Delta/2)^{i-1}}}{\Delta(\Delta/2)^{i-1}} + \sum_{i=k+1}^{depth_{IF}} \frac{m}{(\Delta/2)^{i}} \right\}. \tag{19}$$

Following similar steps to the ones used in *Proposition 5*, it is possible to show that the minimum *size* is obtained for very small *fan-ins* $\Delta_{optim} = 3\ldots6$.

Based on closer estimates of *area* and *delay*, results have also been proved for VLSI-efficient implementations of $IF_{n,m}$ functions (Beiu 1997a, 1998b).

Proposition 7 (Beiu 1997a) *The VLSI-optimal NN which computes the COMPARISON of two n-bit numbers has small-constant fan-in 'neurons' with small-constant bounded weights and thresholds.*

The minimum AT^2 is obtained for $\Delta_{optim} = 6\ldots9$ (as the proof has been obtained using several approximations: neglecting ceilings, using the complexity estimate, etc.). This result has been extended to $IF_{n,m}$ functions. We mention that there are similar *small constants* relating to our ***capacity of processing information*** (Miller, 1956). If a three dimensional hardware implementation would be possible, the energy (*i.e.*, VT^2) will be minimised for $\Delta = 36\ldots81$, which is still small (as opposed to the *fan-in* of the nervous cells in the brain which is normally in the range $10^3\ldots10^4$).

A completely different approach is to use Kolmogorov's superpositions, which shows that there are NNs having only $2n + 1$ neurons (*i.e.*, *size*-optimal) which can approximate any function. We start from a constructive solution for the general case (Sprecher, 1996a, 1996b, 1997).

Proposition 8 (Sprecher, 1996a) *Define the function* $\psi : \mathcal{E} \to \mathcal{D}$ *such that for each integer* $k \in N$:

$$\psi\left(\sum_{r=1}^{k} i_r \gamma^{-r}\right) = \sum_{r=1}^{k} \tilde{i}_r 2^{-m_r} \gamma^{-\frac{n^{r-m_r}-1}{n-1}} \tag{20}$$

where $\tilde{i}_r = i_r - (\gamma - 2)\langle i_r \rangle$ *and*

$$m_r = \langle i_r \rangle \times \left\{ 1 + \sum_{s=1}^{r-1} [i_s] \times \ldots \times [i_{r-1}] \right\} \tag{21}$$

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

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

If we limit the functions to BFs, one digit ($k = 1$) is enough, which gives $\psi(0.i_1) = 0.i_1$, *i.e.* the identity function $\psi(x) = x$. Such a solution builds simple analog neurons having *fan-in* $\Delta \le 2n + 1$. The known *weight* bounds (holding for $\Delta \ge 4$) are (Myhill & Kautz, 1961; Raghavan, 1988; Parberry, 1994; Sontag, 1996):

$$2^{(\Delta-1)/2} < weight < (\Delta + 1)^{(\Delta+1)/2} / 2^{\Delta}. \tag{22}$$

Thus, a *precision* of between Δ, and $\Delta \log\Delta$ bits per *weight* would be expected. Unfortunately, the constructive solution for Kolmogorov's superpositions requires a double exponential precision for ψ (eq. 20), and for the *weights*:

$$\alpha_p = \sum_{r=1}^{\infty} \gamma^{-(p-1)\frac{n^r-1}{n-1}}. \tag{23}$$

For BFs precision is reduced to $(2n + 2)^{-n}$, or $2n\log n$ bits per *weight*. Analog implementations are limited to just several bits of precision (Kramer, 1996), this being one of the reasons for investigations on precision (Denker & Wittner, 1988; Holt & Hwang, 1993; Wray & Green, 1995; Stevenson & Huq, 1996), and on algorithms relying on limited

integer *weights* (Khan & Hines, 1994; Drăghici & Sethi, 1997; Beiu, 1998a). Due to the limitation on precision the solution for implementing BFs should decompose the given BF in simpler BFs which can be efficiently implemented based on Kolmogorov's superpositions (*i.e.*, we have to reduce n to small values). The partial results from this first layer of analog building blocks can be combined using again Kolmogorov's superpositions. The final analog implementation will requires more than three layers. It follows that a systematic solution which would utilise silicon to the best advantage would be to rewrite a given computation (*i.e.*, set of BFs) in a base larger than 2 (*e.g.*, in base 4 as in the previous example), and use Kolmogorov's superpositions for analog implementation of the digit-wise computations in this larger base.

4. Conclusions

The main conclusion is that hardware implementations of NNs are highly limited by the two dimensional mapping into silicon, which leads to very limited *fan-in* and precision. For example, arbitrary BFs can be implemented using:

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

Clearly, there are interesting *fan-in* dependent *depth-size* and *area-delay* tradeoffs. Even more, there are optimal solutions having small constant *fan-in* values, and the problem is not alleviated by futuristic three dimensional optical implementations.

These results also suggest that:

- the brain does not optimise energy and power—like engineers do when designing integrated circuits—and might trade-off the slower individual speeds of its elementary computing elements (thus, reducing power), for their higher connectivity (larger *fan-ins*);
- two dimensional silicon implementations are limited with respect to connectivity, and might only slightly compensate by using higher computing speeds (see *Figure 1.a*);
- three dimensional hardware implementations (*e.g.*, optical) might be still lagging behind biological ones with respect to connectivity, but it is to be expect that the higher computing speed might eventually compensate for that.

Future work should concentrate on finding closer estimates for analog / digital as well as optical implementations.

References

Abu-Mostafa, Y.S. (1988). Connectivity Versus Entropy. In D.Z. Anderson (ed.), *Neural Info. Proc. Sys.*, 1-8. New York, NY: AIPress.

Arai, M. (1993). Bounds on the Number of Hidden Units in Binary-valued Three-layer Neural Networks. *Neural Networks*, **6**(6):855-860.

Arbib, M.A. (1995). *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA: MIT Press.

Attali, J.-G. & Pagès, G. (1997). Approximations of Functions by a Multilayer Perceptron: a New Approach. *Neural Networks*, **10**(6):1069-1081.

Barron, A.R. (1993). Universal Approximation Bounds for Superpositions of a Sigmoidal Function. *IEEE Trans. Info. Theory*, **39**(3):930-945.

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 as: (1997), Valid Generalization, the Size of the Weights is More Important than the Size of the Network, in M.C. Mozer, M.I. Jordan & T. Petsche (eds.): *Advs. in Neural Info. Proc. Sys.*, Cambridge, MA: MIT Press).

Baum, E.B. (1988). On the Capabilities of Multilayer Perceptrons. *J. Complexity*, **4**:193-215.

Beiu, V. (1996a). Entropy Bounds for Classification Algorithms. *Neural Network World*, **6**(4):497-505.

Beiu, V. (1996b). Digital Integrated Circuit Implementations. *Chapter E1.4* in (Fiesler & Beale, 1996).

Beiu V. (1997a). Constant Fan-In Digital Neural Networks Are VLSI-Optimal. *Chapter 12* in S.W. Ellacott, J.C. Mason & I.J. Anderson (eds.), *Mathematics of Neural Networks: Models, Algorithms and Applications*, 89-94, Boston, MA: Kluwer Academic.

Beiu, V. (1997b). When Constants Are Important. In I. Dumitrache (ed.), *Proc. Control Sys. & Comp. Sci. CSCS-11*, vol. **2**, 106-111, Bucharest, Romînia: UPB Press.

Beiu, V. (1998a). Reduced Complexity Constructive Learning Algorithm. *Neural Network World*, **8**(1):1-38.

Beiu, V. (1998b). On the Circuit and VLSI Complexity of Threshold Gate COMPARISON. *Neurocomputing*, **19**(1): 77-98.

Beiu, V. (1998c). *VLSI Complexity of Discrete Neural Networks*. Newark, NJ: Gordon & Breach (to appear).

Beiu, V. & De Pauw, T. (1997). Tight Bounds on the Size of Neural Networks for Classification Problems. In J. Mira, R. Moreno-Díaz & J. Cabestany (eds.): *Biological and Artificial Computation*, 743-752, Berlin: Springer-Verlag.

Beiu, V. & Drăghici, S. (1997). Limited Weights Neural Networks: Very Tight Entropy Based Bounds. In D.W. Pearson (ed.): *Proc. ICSC Symp. on Soft Computing*, 111-118, Millet, Canada: ICSC Acad. Press.

Beiu, V. & Makaruk, H.E. (1998). Deeper Sparser Nets Can Be Optimal. *Neural Proc. Lett.* (to appear).

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

Beiu, V., Peperstraete, J.A., Vandewalle, J. & Lauwereins, R. (1994). Area-Time Performances of Some Neural Computations. In P. Borne, T. Fukuda & S.G. Tzafestas (eds.), *IMACS Intl. Symp. on Signal Proc., Robotics, and Neural Networks SPRANN'94*, 664-668, Lille, France: GERF EC.

Beiu, V., Drăghici, S. & De Pauw, T. (1998). A Constructive Approach to Calculating Lower Entropy Bounds. *Neural Proc. Lett.* (to appear).

Blum, E. & Li, K. (1991). Approximation Theory and Feedforward Networks. *Neural Networks*, **4**(3):511-515.

Bruck, J. & Goodmann, J.W. (1988). On the Power of Neural Networks for Solving Hard Problems. In D.Z. Anderson (ed.), *Neural Info. Proc. Sys.*, 137-143. New York, NY: AIPress (also as (1990), *J. Complexity* **6**:129-135).

Bulsari, A. (1993). Some Analytical Solutions to the General Approximation Problem for Feedforward Neural Networks. *Neural Networks*, **6**(7):991-996.

Cybenko, G. (1988). Continuous Valued Neural Networks with Two Hidden Layers Are Sufficient. *Tech. Rep.*, Tufts Univ., Medford, MA.

Cybenko, G. (1989). Approximations by Superpositions of a Sigmoid Function. *Math. of Control, Signals and Systems*, **2**:303-314.

Denker, J.S. & Wittner, B.S. (1988). Network Generality, Training Required, and Precision Required. In D.Z. Anderson (ed.): *Neural Info. Proc. Sys.*, 219-222, New York, NY: AIPress.

Drăghici, S. & Sethi, I.K. (1997). 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*, 753-762, Berlin: Springer-Verlag.

Fiesler, E. & Beale, R. (1996). *Handbook of Neural Computation*. New York, NY: Oxford Univ. Press & the Inst. of Physics.

Funahashi, K.-I. (1989). On the Approximate Realization of Continuous Mapping by Neural Networks. *Neural Networks*, **2**(2):183-192.

Funahashi, K.-I. & Nakamura, Y. (1993). Approximation of Dynamical Systems by Continuous Time Recurrent Neural Networks. *Neural Networks*, **6**(6):801-806.

Geva, S. & Sitte, J. (1992). A Constructive Method for Multivariate Function Approximation by Multilayered Perceptrons. *IEEE Trans. Neural Networks*, **3**(4):621-623.

Glesner, M. & Pöchmüller, W. (1994). *Neurocomputers – An Overview of Neural Networks in VLSI*. London, UK: Chapman & Hall.

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

Hartman, E., Keeler, J.D. & Kowalski, J.M. (1989). Layered Neural Networks with Gaussian Hidden Units as Universal Approximations. *Neural Computation*, **2**(2): 210-215.

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

Hecht-Nielsen, R. (1987). Kolmogorov's Mapping Neural Network Existence Theorem. *Proc. Intl. Conf. on Neural Networks*, 11-14, Los Alamitos, CA: IEEE CS Press.

Holt, J.L. & Hwang, J.-N. (1993). Finite Precision Error Analysis of Neural Network Hardware Implementations. *IEEE Trans. Comp.*, **42**(3): 281-290.

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

Hornik, K. (1991). Approximation Capabilities of Multilayer Feedforward Networks. *Neural Networks*, **4**(2): 251-257.

Hornik, K. (1993). Some New Results on Neural Network Approximation. *Neural Networks*, **6**(8):1069-1072.

Hornik, K., Stinchcombe M. & White, H. (1989). Multilayer Feedforward Neural Networks Are Universal Approximators. *Neural Networks*, **2**(3):359-366.

Hornik, K., Stinchcombe M. & White, H. (1990). Universal Approximation of an Unknown Mapping and Its Derivatives Using Multilayer Feedforward Networks. *Neural Networks*, **3**(4):551-560.

Huang, S.-C. & Huang, Y.-F. (1991). Bounds on the Number of Hidden Neurons of Multilayer Perceptrons in Classification and Recognition. *IEEE Trans. Neural Networks*, **2**(1):47-55.

Irie, B. & Miyake, S. (1988). Capabilities of Three-Layered Perceptrons. *Proc. Intl. Conf. on Neural Networks*, vol. **1**, 641-648, Los Alamitos, CA: IEEE CS Press.

Ito, Y. (1991). Approximation of Functions on a Compact Set by Finite Sums of Sigmoid Functions without Scaling. *Neural Networks*, **4**(7):817-826.

Ito, Y. (1994). Approximation Capabilities of Layered Neural Networks with Sigmoid Units on Two Layers. *Neural Computation*, **6**(6):1233-1243.

Jones, L.K. (1992). A Simple Lemma on Greedy Approximation in Hilbert Space and Convergence Rates for Projection Pursuit Regression and Neural Network Training. *Ann. Statist.*, **20**:608-613.

Katsuura, H. & Sprecher, D.A. (1994). Computational Aspects of Kolmogorov's Superposition Theorem. *Neural Networks*, **7**(3):455-461.

Khan, A.H. & Hines, E.L. (1994). Integer-Weight Neural Networks. *Electr. Lett.*, **30**(15):1237-1238.

Koiran, P. (1993). On the Complexity of Approximating Mappings Using Feedforward Networks. *Neural Networks*, **6**(5):649-653.

Kolmogorov, A.N. (1957). On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition. *Dokl. Akad. Nauk SSSR*, **114**:953-956 (English transl. (1963). *Transl. Amer. Math. Soc.*, **2**(28):55-59).

Kramer, A.H. (1996). Array-Based Analog Computation; Principles, Advantages and Limitations. *Proc. Microelectronics for Neural Networks*, 68-79, Los Alamitos, CA: IEEE CS Press.

Kůrková, V. (1992). Kolmogorov's Theorem and Multilayer Neural Networks. *Neural Networks*, **5**(4):501-506.

Kůrková, V., Kainen, P.C. & Kreinovich, V. (1997). Estimates of the Number of Hidden Units and Variations with Respect to Half-Spaces. *Neural Networks*, **10**(6): 1061-1068.

LeCun, Y. (1987). Models connexionistes de l'apprentisage. *MSc thesis*, Univ. Pierre et Marie Curie, Paris.

Leshno, M., Lin, V.Y., Pinkus, A. & Schocken, S. (1993). Multilayer Feedforward Neural Networks with a Nonpolynomial Activation Function Can Approximate any Function. *Neural Networks*, **6**(6):861-867.

Lippmann, R.P. (1987). An Introduction to Computing with Neural Nets. *IEEE ASSP Mag.*, **4**(2):4-22.

Lupanov, O.B. (1973). The Synthesis of Circuits from Threshold Elements. *Problemy Kibernetiki*, **20**:109-140.

Mhaskar, H.N. & Micchelli, C. (1992). Approximation by Superposition of Sigmoidal and Radial Basis Functions. *Advs. in Appl. Maths.*, **13**: 350-373.

Mhaskar, H.N. & Micchelli, C. (1994). Dimension Independent Bounds on the degree of Approximation by Neural Networks. *IBM J. Res. and Dev.*, **38**(3):277-283.

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.

Minnik, R.C. (1961). Linear-Input Logic. *IRE Trans. Electr. Comp.*, **10**:6-16.

Myhill, J. & Kautz, W.H. (1961). On the Size of Weights Required for Linear-Input Switching Functions. *IRE Trans. Electr. Comp.*, **10**:288-290.

Neciporuk, E.I. (1964). The Synthesis of Networks from Threshold Elements. *Soviet Mathematics—Doklady*, **5**(1):163-166 (English transl. (1964), *Automation Express*, 7(1):35-39 & 7(2):27-32).

Nees, M. (1994). Approximate Versions of Kolmogorov's Superposition Theorem, Proved Constructively. *J. Comp. and Appl. Math.*, **54**(2):239-250.

Nees, M. (1996). Chebyshev Approximation by Discrete Superposition. Application to Neural Networks. *Advs. in Comp. Maths.*, **5**(2):137-152.

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

Park, J. & Sandberg, I.W. (1991). Universal Approximation Using Radial-Basis-Function Networks. *Neural Computation*, **3**(2):246-257.

Park, J. & Sandberg, I.W. (1993). Approximation and Radial-Basis-Function Networks. *Neural Computation*, **5**(3):305-316.

Paugam-Moisy, H. (1992). Optimisation des réseaux des neurones artificiels. *PhD Dissertation*, LIP (http://www.ens-lyon.fr/LIP/publis.us.html), École Normale Supérieure de Lyon, 46 Allée d'Italie, 69364 Lyon, France.

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

Poggio, T. & Girosi, F. (1989). A Theory of Networks for Approximation and Learning. *Tech. Rep. AI Memo 1140* (ftp://publications.ai.mit.edu/ai-publications/1000-1499/AIM-1140.ps.Z), MIT, MA, 87 pages (short version as (1990), Networks for Approximation and Learning, *Proc. IEEE*, **78**(9): 1481-1497).

Raghavan, P. (1988). Learning in Threshold Networks: A Computational Model and Applications. *Tech. Rep. RC 13859*, IBM Res. (also in (1988), *Proc. Comp. Learning Theory*, 19-27, New York, NY: ACM Press).

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

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

Scarselli, F. & Tsoi, A.C. (1998). Universal Approximation Using Feedforward Neural Networks: A Survey of Some Existing Methods, and Some New Results. *Neural Networks*, **11**(1):15-37.

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

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

Sontag, E.D. (1996). Shattering All Sets of k Points in "General Position" Requires $(k-1)/2$ Parameters. *Report SYCON 96-01*, Maths. Dept., Rutgers Univ. (also as (1997), *Neural Computation*, **9**(2):337-348).

Sprecher, D.A. (1965). On the Structure of Continuous Functions of Several Variables. *Trans. American Math. Soc.*, **115**:340-355.

Sprecher, D.A. (1966). On the Structure of Representations of Continuous functions of Several Variables as Finite Sums of Continuous Functions of One Variable. *Proc. American Math. Soc.*, **17**:98-105.

Sprecher, D.A. (1993). A Universal Mapping for Kolmogorov's Superposition Theorem. *Neural Networks*, **6**(8):1089-1094.

Sprecher, D.A. (1996a). A Numerical Implementation of Kolmogorov's Superpositions. *Neural Networks*, **9**(5): 765-772.

Sprecher, D.A. (1996b). A Numerical Construction of a Universal Function for Kolmogorov's Superpositions. *Neural Network World*, **6**(4): 711-718.

Sprecher, D.A. (1997). A Numerical Implementation of Kolmogorov's Superpositions II. *Neural Networks*, **10**(3):447-457.

Stevenson, M. & Huq, S. (1996). On the Capability of Threshold Adalines with Limited-Precision Weights. *Neural Computation*, **8**(8):1603-1610.

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

Williamson, R.C. (1990). ε-Entropy and the Complexity of Feedforward Neural Networks. In R.P. Lippmann, J.E. Moody & D.S. Touretzky (eds.), *Advs. in Neural Info. Proc. Sys.*, 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 734*, Schloß Birlinghoven, St. Augustin, Germany (also as (1993), *Complex Systems*, 7(3):199-220).