DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
DISCLAIMER

 Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.
ON ANALOG IMPLEMENTATION OF DISCRETE NEURAL NETWORKS

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

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

Keywords—neural networks, Kolmogorov's superimpositions, threshold gate circuits, analog circuits, size, precision.

1 Introduction

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

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

---

\(^{\text{On leave of absence from the "Politehnica" University of Bucharest, Computer Science Department, Spl. Independenței 313, RO-77206 Bucharest, România.}}\)
2 Previous Results

NNs have been experimentally shown to be quite effective in many applications (see Applications of Neural Networks\(^1\), together with Part F: Applications of Neural Computation\(^2\) and Part G: Neural Networks in Practice: Case Studies\(^3\)). This success has led researchers to undertake a rigorous analysis of the mathematical properties that enable them to perform so well, and has generated two directions of research: (i) to find existence/constructive proofs for what is now known as the “universal approximation problem”; (ii) to find tight bounds on the size needed by the approximation problem (or some particular cases). The paper will focus on both aspects, for the particular case when the functions to be implemented are BFs.

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

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

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

of Kolmogorov’s superpositions\(^12\):

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

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 has been given the next year by Cybenko\(^13,14\) using a continuous activation function and was independently presented by Irie and Miyake.\(^15\) Thus, the fact that NNs are computationally universal—with more or less restrictive conditions—when modifiable connections are allowed, was established. Different enhancements have been later presented in the literature\(^16,17\)(Chp. 1).

- Funahashi\(^18\) proved the same result in a more constructive way and also refined the use of Kolmogorov’s theorem, giving an approximation result for two-hidden-layer NNs;
- Hornik et al.\(^19\) showed that the continuity requirement for the output function can be partly removed;
- Hornik et al.\(^20\) also proved that a NN can approximate simultaneously a function and its derivative;
- Park and Sandberg\(^21,22\) used radial basis functions in the hidden layer, and gave an ‘almost’ constructive proof;
- Hornik\(^23\) showed that the continuity requirement can be completely removed, the activation function having to be ‘bounded and nonconstant’;
- Geva and Sitte\(^24\) proved that four-layered NNs with sigmoid activation function are universal approximators;
Kůrkova\textsuperscript{25} has demonstrated the existence of approximate superposition representations, \textit{i.e.} \( \psi \) and \( \Phi \), can be approximated with functions of the form \( \sum a_i \sigma(b_i x + c_i) \), where \( \sigma \) is an arbitrary activation sigmoidal function;

- Mhaskar and Micchelli\textsuperscript{26,27} approach was based on the Fourier series of the function, by truncating the infinite sum to a finite set, and rewriting \( e^{ikx} \) in terms of the activation function (which has to be periodic);

- Koiran\textsuperscript{28} presented a new proof on the line of Funahashi's,\textsuperscript{18} but more general in that it allows the use of units with 'piecewise continuous' activation functions; these include the important case of \textit{threshold gates} (TGs);

- Leshno \textit{et al.}\textsuperscript{29} relaxed the condition for the activation function to 'locally bounded piecewise continuous' \textit{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\textsuperscript{30} added to these results by proving that: (i) if the activation function is locally Riemann integrable and nonpolynomial, the \textit{weights} and the \textit{thresholds} can be constrained to arbitrarily small sets; and (ii) if the activation function is locally analytic, a single universal \textit{threshold} will do;

- Funahashi and Nakamura\textsuperscript{6} showed that the universal approximation theorem also holds for trajectories of patterns;

- Sprecher\textsuperscript{31} has demonstrated that there are universal hidden layers that are independent of the number of input variables \( n \);

- Barron\textsuperscript{32} described spaces of functions that can be approximated by Jones' algorithm\textsuperscript{33} using functions computed by single-hidden-layer network of TGs;

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

\subsection*{2.2 Bounds on the Size of Threshold Gate Circuits}

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

\[ \text{size} \geq 2 \left(2^n / n \right)^{1/2}, \]

while later a very tight upper bound has been proven\textsuperscript{42} in \textit{depth} = 4:
A similar existence lower bound for arbitrary BFs is $\Omega(2^{n/3})$, while Roychowdhury et al. details lower bounds for particular BFs.

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

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

Several other existence lower bounds for arbitrary dichotomy are as follows:

- A depth-2 TGC requires at least $m / (n \log(m/n))$ TGs;
- A depth-3 TGC requires at least $2 (m / \log m)^{1/2}$ TGs in each of the two hidden layers (if $m > n^2$);
- An arbitrarily interconnected TGC without feedback needs $(2m / \log m)^{1/2}$ TGs (if $m > n^2$).

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

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

The particular case of BFs has been intensively studied. Some results have been obtained for particular BFs, but a size-optimal result for BFs that have
exactly $m$ groups of ones in their truth table (equivalently, which are defined on the $m$ groups) was detailed by Red’kin.\textsuperscript{53}

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

All the previous mentioned results are valid for unlimited fan-in TGs. Departing from these lines, Horne and Hush\textsuperscript{54} detail a solution for limited fan-in TGs.

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

### 3 Analog Implementation of Boolean Functions

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

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

**Theorem\textsuperscript{37}** Define the function $\psi : \mathbb{D} \rightarrow \mathbb{D}$ such that for each integer $k \in \mathbb{N}$,

$$
\psi\left(\sum_{r=1}^{k} i_r \gamma^{-r}\right) = \sum_{r=1}^{k} i_r 2^{-m_r - 1} \quad \gamma^{-r} - 1
$$

where

$$
\gamma_r = i_r - (\gamma - 2) \langle i_r \rangle
$$

and

$$
m_r = \langle i_r \rangle (1 + \sum_{r=1}^{m_r} [i_r] \times \ldots \times [i_{r-1}])
$$

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

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

For BFs, one digit is enough ($k = 1$), which gives $\psi(0.i) = 0.i$ (or $\psi(x) = x$), and shows that the nonlinearity is the identity function.
Such a solution builds analog neurons having fan-in $\Delta \leq 2n + 1$, for which the known weight bounds\(^{52,58-60}\) (holding for any fan-in $\Delta \geq 4$) are:

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

(9)

Thus, one would expect to have a precision of between $\Delta$ and $\Delta \log \Delta$ bits per weight. Unfortunately, the solution for Kolmogorov’s superposition\(^{37-50}\) requires (in general) a double exponential precision for $\psi$ (Eq. 6), and for the weights:

$$\alpha_p = \sum_{\gamma=1}^{\infty} \gamma^{-(\gamma-1)} n^{-1}.$$  

(10)

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

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

4 Conclusions

Arbitrary BFs can be implemented using:

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

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

References

17. V. Béiu, VLSI Complexity of Discrete Neural Networks (Gordon and Breach, Newark, NJ, 1998).
34. H. Katsuura and D.A. Sprecher, Neural Networks 7, 455 (1994).
42. O.B. Lupanov, Problemy Kibernetiki 20, 109 (1973).
51. A. Bulsari, Neural Networks 6, 991 (1993).
64. J. Wray and G.G.R. Green, Neural Networks 8, 31 (1995).