HYPERBOLIC SIGMA-PI NEURAL NETWORK OPERATORS APPLIED TO IMAGE PROCESSING AND RECONSTRUCTION

Burkhard Lenze
Fachhochschule Dortmund, Fachbereich Informatik
P.O. Box 105018, D-44047 Dortmund, Germany

Communicated by A. Dishliev

ABSTRACT: In this paper, we recall how to design three-layer feedforward neural network operators based on hyperbolic sigma-pi units in order to act as approximation and interpolation devices for regular gridded data. The basic part of the paper consists of an application of this strategy in connection with image processing and reconstruction.

AMS (MOS) Subject Classification. 26E40, 41A05, 41A63, 92B20

1. INTRODUCTION

As it is well-known, there is a quite intimate connection between approximation and interpolation theory and neural network design. Without any claim of completeness we mention the papers K. Hornik et al [2], C.K. Chui and X. Li [1], B. Lenze [4], H.N. Mhaskar and C.A. Micchelli [9], A. Pinkus [10]. In the following, we recall some special well-established technique from approximation and interpolation theory in order to construct three-layer feedforward networks for discrete data processing and apply these networks resp. operators in connection with image processing and reconstruction. The general idea is to sample the given discrete information on a coarse regular grid and to use this partial information to initialize operators which approximate resp. interpolate the data and yield reasonable results at non-sampling points. Moreover, the resulting operators can be easily implemented in the framework of a three-layer feedforward neural network with hyperbolic sigma-pi units in the hidden layer (cf. B. Lenze [4], B. Lenze [5], B. Lenze [7] for details concerning such networks). Summing up, we obtain three-layer feedforward sigma-pi neural networks of hyperbolic type for approximation and interpolation of regular gridded data applicable in connection with image processing and reconstruction. At the end of this brief
introduction, let us add a few words about the organization of the paper. We start
with some notational preliminaries and introduce the concept of sigmoidal functions
for designing neural network operators. Then, we recall the definition of our basic
operators which yield approximations or interpolations with respect to given regular
discrete information. In the main part of the paper, we discuss an application of the
network operators in the field of image processing and reconstruction.

2. NOTATION AND OPERATORS

We essentially proceed as in B. Lenze [7] and recall the construction of the
operators in order to make the paper self-contained. Let $n \in \mathbb{N}$ be given and
$a = (a_1, a_2, \ldots, a_n), b = (b_1, b_2, \ldots, b_n) \in \mathbb{R}^n$, with $a < b$ (i.e., $a_k < b_k$, $1 \leq k \leq n$)
the endpoints of the interval $[a, b] \subset \mathbb{R}^n$,

$$[a, b] := \{x \in \mathbb{R}^n \mid a_k \leq x_k \leq b_k, \ 1 \leq k \leq n\}. \quad (1)$$

By means of a standard translation argument we may assume that the point
$a \in \mathbb{R}^n$ is always equal to the origin, i.e., without loss of generality we have $a = 0$.
We choose $J_1, J_2, \ldots, J_n \in \mathbb{N}$ and define $h \in \mathbb{R}^n$ componentwise by

$$h_k := \frac{b_k}{J_k}, \quad 1 \leq k \leq n. \quad (2)$$

On the interval $[0, b]$ we introduce the regular grid with grid points

$$h_j := (h_{1j_1}, h_{2j_2}, \ldots, h_{nj_n}) \in [0, b], \quad 0 \leq j \leq J,$$

where $j := (j_1, j_2, \ldots, j_n)$ and $J := (J_1, J_2, \ldots, J_n)$. Moreover, we set $h := he$ with $e := (1, 1, \ldots, 1)$ and call the grid defined in (3) a $(J_1 \times J_2 \times \cdots \times J_n)$-grid on $[0, b]$. At each grid point $h_j$, $0 \leq j \leq J$, there may be given a real value $f(h_j) \in \mathbb{R}$
which we may assume to come from some underlying function $f : [0, b] \rightarrow \mathbb{R}$. Our
problem is to design neural network operators which are able to model the given
discrete information $(h_j, f(h_j)), 0 \leq j \leq J$. To solve this problem, we first of all
need some further definitions which in part may already be found in B. Lenze [4]. For
arbitrary $c, d \in \mathbb{R}^n$, with $c < d$, let

$$Cor[c, d] := \{x \in \mathbb{R}^n \mid x_k = c_k \text{ or } x_k = d_k, \ 1 \leq k \leq n\} \quad (4)$$

be the set of corners of the interval $[c, d]$. Moreover, let

$$\gamma(x, c) := \#\{k \in \{1, 2, \ldots, n\} \mid x_k = c_k\}, \quad x \in Cor[c, d]. \quad (5)$$

In (5), $\#$ denotes the number of distinct elements of the set under consideration
and $n - \gamma(x, c)$ is nothing else but the well-known Hamming distance from $x$ to $c$. 
Now, for a given function \( f : [c, d] \to \mathbb{R} \) the so-called corresponding interval function or iterated difference \( \Delta f \) of \( f \) on \([c, d]\) is defined by
\[
\Delta f[c, d] := \sum_{x \in Cor[c, d]} (-1)^{\gamma(x,c)} f(x).
\] (6)

As the last of these basic definitions we introduce the notion of a sigmoidal function, which is a bounded measurable function \( \sigma : \mathbb{R} \to \mathbb{R} \) satisfying
\[
\lim_{\xi \to -\infty} \sigma(\xi) = 0 \quad \text{and} \quad \lim_{\xi \to \infty} \sigma(\xi) = 1.
\] (7)

Examples of sigmoidal functions are the unit step function \( 1 : \mathbb{R} \to \{0, 1\} \),
\[
1(\xi) := \begin{cases} 
0, & \xi < 0, \\
1, & \xi \geq 0,
\end{cases}
\] (8)
or scaled integrated B-splines as discussed in Section 3. Now, we are prepared to introduce the hyperbolic approximation operators \( \Omega^{(h,\beta)} \). For \( \sigma \) a given sigmoidal function and \( \beta > 0 \) a free dilation parameter, \( e = (1, 1, \ldots, 1) \in \mathbb{Z}^n \), \( h \in \mathbb{R}^n \) given by (2), and \( f(h_j) \in \mathbb{R} \), \( 0 \leq j \leq J \), the given discrete data set induced by some underlying function \( f : [0, b] \to \mathbb{R} \), the operators \( \Omega^{(h,\beta)} \) are defined for all \( x \in \mathbb{R}^n \) as
\[
\Omega^{(h,\beta)}(f)(x) := 2^{1-n} \sum_{\substack{j \in \mathbb{Z}^n \\
eq e \leq j \leq J}} \sigma \left( \beta \prod_{k=1}^{n} \left( \frac{x_k}{h_k} - \frac{j_k}{2} \right) \right) \Delta f[h_j, h_{j+e}].
\] (9)

where we agree to set
\[
f(h_j) := 0, \quad \text{for } h_j \notin [0, b],
\] (10)
in order to obtain the compact notation of the finite sums appearing in (9). Obviously, the operators are basically induced by the given sigmoidal function \( \sigma \) evaluated at componentwise products of shifted and scaled arguments, so-called hyperbolic-type arguments, and the interval function \( \Delta f \) which makes use of all underlying discrete information. It can be shown that the operators are linear and that they approximate \( f \) on the underlying grid or even interpolate \( f \) on the grid in case that \( \sigma(\xi) = 1(\xi) \) for \( |\xi| \geq M \) and \( \beta \geq 2^n M \) (cf. B. Lenze [3]-[8]) for proofs and more details concerning the theory of these operators and their connection to neural networks).

In this contribution, however, we are not interested in abstract properties of the operators defined in (9), but in concrete applications of them in the field of image processing and reconstruction.

3. APPLICATION
As already stated, we will now apply our network operators to a concrete image processing and reconstruction problem. Therefore, we first of all have to answer the question of which sigmoidal functions we should use in order to generate our operators, resp., networks. Because of the essential conditions (7) a nice way to get sigmoidal functions is to start with the well-known B-splines (cf. L. L. Schumaker [11], for example) and integrate them.

\[
B(\xi) := \begin{cases} 1 - |\xi|, & |\xi| \leq 1, \\ 0, & |\xi| > 1, \end{cases}
\]

(11)

and integrate it,

\[
\sigma_B(\xi) := \begin{cases} 0, & \xi \leq -1, \\ \frac{1}{2}\xi^2 + \xi + \frac{1}{2}, & -1 < \xi \leq 0, \\ -\frac{1}{2}\xi^2 + \xi + \frac{1}{2}, & 0 < \xi \leq 1, \\ 1, & \xi > 1. \end{cases}
\]

(12)

The resulting function \(\sigma_B\) is obviously a sigmoidal function. Moreover, this sigmoidal function is monotone increasing, continuously differentiable and identical with the unit step function \(1\) outside the interval \([-1, 1]\). In the following, we will consider the special case \(n = 2\) and only use the scaling factors \(\beta = 1, 2, 4, 10\) (non-interpolation case and smoothing effect for \(\beta = 1, 2\); interpolation case and rough reconstruction for \(\beta = 4, 10\)). In detail, our operators are

\[
\Omega^{(h, \beta)}(f)(x) := \frac{1}{2} \sum_{j \in \mathbb{Z}^2} \sigma_B \left( \frac{\beta}{2} \prod_{k=1}^{2} \left( \frac{x_k}{h_k} - j_k - \frac{1}{2} \right) \right) \Delta_f \left[ h, h + e \right]
\]

(13)
with $e = (1, 1) \in \mathbb{Z}^2$. \( \Delta f \left[ h_j, h_j + e \right] = f(h_j) - f(h_{j+(1,0)}) - f(h_{j+(0,1)}) + f(h_j + e) \), 
\( \beta = 1, 2, 4, 10 \), and \( x \in \mathbb{R}^2 \). To get an idea of the kind of application we have in mind we take a look at Figure 1.

Here, we sample on a coarse \((3 \times 3)\)-grid and reconstruct on a refined \((6 \times 6)\)-grid \((h > 0)\) may be an arbitrary real number, \( h_j := h\) and \([0, b] := [0, 3he] \); moreover \( \beta = 1 \) and \( \sigma = \sigma_B \). It should be noticed that the hidden neurons do not depend on the sampled information but only on the geometry of the sampling and the reconstruction grid. Therefore, the hidden neurons resp. the arguments of the sigmoidal function can be fixed in advance and all information for the whole set of pixel points in the reconstructed image can be obtained in parallel using this fixed configuration of the hidden neurons. The sampled data is only used to initialize the output weights resp. the functionals \( \Delta f \) and do not affect the hidden neurons. These facts are essential in order to guarantee a fast and parallel image processing in case that not only one picture but a sequence of pictures has to be transferred. In the following, we consider an example where the sampling grid is \((20 \times 20)\) and the reconstruction grid is \((80 \times 80)\).

As a test function or even better as a test image for our operators we take a look at a problem to sample and reconstruct the capital letter “O”. First of all, we define this letter on the interval \([0, 1] \times [0, 1]\) using the following function \( f : [0, 1]^2 \rightarrow \mathbb{R} \),

\[
f(x_1, x_2) := \begin{cases} 
1, & \text{if } \frac{1}{20} < 2|x_1 - \frac{1}{2}|^2 + |x_2 - \frac{1}{2}|^2 < \frac{1}{5}, \\
0, & \text{elsewhere.} 
\end{cases}
\]

(14)

We set \([0, b] := [0, 1] \times [0, 1], \beta = 1, 2, 4, 10, \mathbf{J} := (20, 20)\), and \( \mathbf{h} := (\frac{1}{20}, \frac{1}{20}) = \frac{1}{20}e \), and obtain the following contour plots.
Figure 2: contour plot of $f$ on a $(80 \times 80)$-grid

Figure 3: contour plot of the sampling of $f$ on a $(20 \times 20)$-grid
Figure 4: contour plot of $\Omega(\frac{1}{20}e^{10}) (f)$ on a $(80 \times 80)$-grid

Figure 5: contour plot of $\Omega(\frac{1}{20}e^4) (f)$ on a $(80 \times 80)$-grid
Figure 6: contour plot of $\Omega(\frac{1}{20}e, 2)(f)$ on a $(80 \times 80)$-grid

Figure 7: contour plot of $\Omega(\frac{1}{20}e, 1)(f)$ on a $(80 \times 80)$-grid
Figure 2 shows the original image “O” on a $(80 \times 80)$-grid and Figure 3 its sampling on a $(20 \times 20)$-grid. Based on this partial information the operators try to reconstruct the image on the original $(80 \times 80)$-grid. The result for the scaling parameter $\beta = 10$ as visualized in Figure 4 shows that in this case the reconstruction is rather proper and the edges are rough. Moreover, for this case it is well-known that the operators even interpolate the sampled information on the initial coarse grid. This is still true for $\beta = 4$, but for $\beta < 4$ interpolation can no longer be guaranteed. Moreover, when decreasing $\beta$ as in Figures 5-7 we can recognize a blur effect: sharp edges become smoother and smoother and different local information is averaged in order to use more and more global values for local reconstruction. As it is well-known, blur techniques of this type are sometimes highly appreciated in image processing and manipulation.

REFERENCES


