

Wavelet Neural Networks Are Asymptotically Optimal Approximators For Functions of One Variable

Vladik Kreinovich¹, Ongard Sirisaengtaksin^{1,2}, and Sergio Cabrera³

Abstract— Neural networks are universal approximators. For example, it has been proved (Hornik et al) that for every $\varepsilon > 0$, an arbitrary continuous function on a compact set can be ε -approximated by a 3-layer neural network. This and other results prove that in principle, any function (e.g., any control) can be implemented by an appropriate neural network. But why neural networks? In addition to neural networks, an arbitrary continuous function can be also approximated by polynomials, etc. What is so special about neural networks that make them preferable approximators?

To compare different approximators, one can compare the number of bits that we must store in order to be able to reconstruct a function with a given precision ε . For neural networks, we must store weights and thresholds. For polynomials, we must store coefficients, etc. In the present paper, we consider functions of one variable, and show that for some special neurons (corresponding to wavelets), neural networks are optimal approximators in the sense that they require (asymptotically) the smallest possible number of bits.

I. BRIEF INFORMAL INTRODUCTION

Recently, it has been proved that neurons are universal approximators (see, e.g., [11], [6], [10], [13], [3], [1], [12]). These results prove that in principle, any function (e.g., any control) can be implemented by an appropriate neural network. But why neural networks? In addition to neural networks, an arbitrary continuous function can be also approximated by polynomials, etc. Is there anything special in

The authors are with the ¹Computer Science and ³Electrical Engineering Departments of the University of Texas at El Paso, El Paso, TX 79968, and with the ²Department of Applied Mathematical Sciences, University of Houston, Downtown, Houston, TX 77002. This work was supported in part by NSF Grant No. CDA-9015006, NASA Research Grant No. 9-482, and a Grant No. PF90-018 from the General Services Administration (GSA) administered by the Institute for Materials and Manufacturing Management.

neural networks that make them preferable approximators?

To compare different approximators, one can compare the number of bits that we must store in order to be able to reconstruct a function with a given precision ε . For neural networks, we must store weights and thresholds. For polynomials, we must store coefficients, etc. In the present paper, we consider functions of one variable, and show that for some special neurons (in which activation functions correspond to wavelets), neural networks are optimal approximators in the sense that they require (asymptotically) the smallest possible number of bits.

The structure of this paper is as follows. Our main goal is to prove that some special type of feedforward neural networks (namely, wavelet neural networks), are the best of all possible approximators. The more general is our definition of an approximator, the stronger the result. Therefore, in Section II, we choose a reasonable class of functions $f(x)$, and for this class, describe the most general definition of an approximation procedure. In Section III, we describe wavelet neural networks. Our main result is presented in Section IV.

II. APPROXIMATION PROCEDURES: BASIC DEFINITIONS AND THEIR MOTIVATIONS

Let us first describe the class of functions $y = f(x)$ that we want to approximate, explain why we have chosen this class, and give examples of reasonable approximators.

Definition 1. Suppose that p is a positive integer, and $\Delta > 0$ is a real number. By $F_p(\Delta)$ we denote the set of all functions $f(x)$ that have p continuous derivatives on $(0,1)$, are equal to 0 in some neighborhoods of 0 and 1, and satisfy the inequality $|f^{(p)}(x)| \leq \Delta$ for all $x \in (0,1)$.

Motivation. In the present paper, we will analyze a 1-dimensional approximation problem. By this, we mean the following: suppose that we have a physical dependency $f(x)$ that is a smooth function. By *smooth* we mean that $f(x)$ is a p times differentiable function of x , for some integer p . We want to

measure the values $f(x_i)$ of this process, and then transform these values into some computer record r in such a way that from r , we can reconstruct the whole function $f(x)$. This record r will be called the *approximator* for $f(x)$, or the *result of compressing* $f(x)$. Let us give two examples of approximations:

Example 1. When we say that 3-layer neural network approximate functions, we mean that an arbitrary function $f(x) : R \rightarrow R$ can be approximated by the expressions of the type

$$\sum_{h=1}^H \beta_h s(w_h x + b_h).$$

Here H is the number of neurons in the hidden layer, w_h is the weight on the way from the input to h -th hidden neuron, b_h is the threshold of h -th hidden neuron, β_h is the weight on the way from h -th hidden neuron to the output neuron, and $s(x)$ is a function that is called an *input-output characteristics* of a neuron. This function is supposed to be fixed. So, in this example, a record r contains the binary expansions of the coefficients β_h , w_h , and b_h .

Example 2. We can also approximate an arbitrary function by polynomials:

$$f(x) \approx \sum_{k=0}^K C_k x^k.$$

Here, a record r consists of the binary codes of the coefficients C_k .

In real life situations, we can only observe the values x_i that do not exceed some limit X . So, if we say that we know the entire dependency, this means that the value $f(x)$ is not physically meaningful for $x > X$ or $x < -X$. For example, if x is velocity, then it makes no sense to ask for $f(x)$ for x greater than the velocity of light c . In view of that, let us assume that the function $f(x)$ is different from 0 only on some finite interval $(-X, X)$. In mathematical terms, we assume that a function $f(x)$ has a finite support.

Of course, the bigger this interval $(-X, X)$, the more sample values we must measure, and therefore, the more information we have to store. So, to compare different approximation schemes, we must fix an interval. In the present paper, we will do all the computations with the interval $(0,1)$. However, by choosing this interval we do not lose any generality: the same asymptotic estimates are true for any interval $(-X, X)$.

We consider smooth dependencies $y = f(x)$ only. The faster y changes with x , the smaller must be the interval between the two consequent values x_i if we

want to monitor all the changes in $y = f(x)$. Therefore, the more values we have to store. Likewise, if we consider functions that are two times differentiable, the bigger is the possible acceleration, the more closely we have to monitor $f(x)$. In view of that, if we consider all possible smooth functions, and do not restrict the derivative, then we will be unable to compare different approximation schemes: for each scheme, the maximum number of bits that is necessary to reconstruct such a function tends to ∞ as the derivative tends to ∞ .

So, to compare different approximation schemes, let us consider only the functions $f(x)$, for which the value of p -th derivative is limited by some number $\Delta > 0$. Thus, we restrict ourselves to the functions from the above-defined class $F_p(\Delta)$.

Denotation. By $\rho(f, g)$ we will denote an L^2 metric on a function space, i.e., $\rho(f, g) = \|f - g\|$, where $\|f - g\|^2 = \int (f(x) - g(x))^2 dx$.

Motivation. As the authors of [9] correctly state, some researchers use the mean-square error (i.e., L^2 -metric) without a priori justification, and in some problems (like image processing) this metric may not give the best description of what we humans mean by close images.

Therefore, let us find out what metric is most adequate to describe the mainstream approximation problems.

Since all the measurements are inevitably imprecise, we cannot reconstruct $f(x)$ precisely. Usually, the measurement errors are independently distributed. So, the differences e_i between the actual values $f(x_i)$ and the measured values $\tilde{f}(x_i)$ are equally normally distributed independent random variables. According to a χ^2 -criterion, we can conclude that with given reliability, the possible values of e_i must satisfy the inequality $\sum_i e_i^2 \leq C$ for some constant C . Therefore, if we have two different functions $f(x)$ and $\tilde{f}(x)$ for which $\sum_i (f(x_i) - \tilde{f}(x_i))^2 \leq C$, and the results of the measurements are $\tilde{f}(x_i)$, then we cannot tell whether the actual function is $\tilde{f}(x)$, or the actual signal is $f(x)$, and the measurement results are different from $f(x_i)$ because of the measurement errors. So, the same measurement results can be due to both functions: $f(x)$ and $\tilde{f}(x)$.

We want to reconstruct the function $f(x)$ as precisely as possible. Therefore, we must take measurements in many points. The expression $\sum_i (f(x_i) - \tilde{f}(x_i))^2$ is an integral sum for the integral $\int (f(x) - \tilde{f}(x))^2 dx$. When we take sufficiently many x_i , we get

$$\sum_i (f(x_i) - \tilde{f}(x_i))^2 \approx K \int (f(x) - \tilde{f}(x))^2 dx,$$

and the inequality $\sum_i (f(x_i) - \tilde{f}(x_i))^2 \leq C$ turns into $\int (f(x) - \tilde{f}(x))^2 dx \leq \alpha$ for some $\alpha > 0$.

Therefore, even if we do not do any approximation and keep all measurement results, we cannot reconstruct a function $f(x)$ precisely: as a result, we can get any function $\tilde{f}(x)$ that satisfies the inequality $\int (f(x) - \tilde{f}(x))^2 dx \leq \alpha$.

The closer α to 0, the more precise is this reconstruction. Therefore, the error of the reconstruction is best expressed by a L^2 metric. So, we can formulate the approximation problem as follows: we want to transform the values $f(x_i)$ into a record r in such a way that from this record r , we will be able to reconstruct a function $\tilde{f}(x)$ that is sufficiently close to $f(x)$ in the sense that

$$\int (f(x) - \tilde{f}(x))^2 dx \leq \varepsilon^2,$$

where we denoted $\varepsilon = \sqrt{\alpha}$. In the above denotations, this inequality can be rewritten as $\rho(f, \tilde{f}) \leq \varepsilon$.

Remark. For specific problems (e.g., in robust control systems design), other metrics are more adequate.

Definition. Suppose that F is a metric space with a metric ρ , and $\varepsilon > 0$. By a ε -net we mean a set $M \subset F$ such that for every $f \in F$, there exists an $m \in M$ such that $\rho(f, m) \leq \varepsilon$.

Now, we are ready to define an approximation set.

Definition 2. For a given $\varepsilon > 0$, by an *approximation set* for a set $F_p(\Delta)$, we mean a finite set M that is an ε -net for $F_p(\Delta)$ (in L^2 -metric). The number ε will be called a *precision* of this approximation set. We will say that an approximation set M requires $\lceil \log_2 |M| \rceil$ bits, where $|M|$ denotes the number of elements in M .

Motivation. Let us first show that any real-life approximation procedure leads to a sequence $\{f_i\} \subset F_p(\Delta)$. Indeed, let us take an arbitrary approximation procedure. Let us denote by B the number of bits that this procedure uses to store the information about a signal. This means that every record is represented by a sequence of B 0's and 1's. Since there are 2^B such sequences, we get at most 2^B different records. Each record can be also interpreted as a binary code of an integer j (so that $0 \leq j \leq 2^B - 1$). Let us denote by $f_i(x)$, $1 \leq i \leq 2^B$, a function that is reconstructed from a record $i-1$. Then, according to our understanding of an approximation problem, every function $f(x)$ from $F_p(\Delta)$ can be transformed into some record j , so that the result f_{j+1} of reconstructing from this record is ε -close to f . So, $\{f_i\}$ form an ε -net.

Let us now explain why any ε -net can be viewed as an approximation set. Indeed, if $\{f_i\}$ is an ε -net, then for each function $f(x)$, there exists an i such that $\rho(f, f_i) \leq \varepsilon$. So, we can use $j = i - 1$ as a record that corresponds to $f(x)$.

In general, if we have $|M|$ different elements in a ε -net, then we can use numbers from 0 to $|M| - 1$ to store them. B bits allow us to store 2^B numbers (from 0 to $2^B - 1$). Therefore, to store numbers from 1 to $|M|$, we need $\lceil \log_2 |M| \rceil$ bits.

Definition 3. By an *approximation scheme* S for a set $F_p(\Delta)$, we mean a function that transforms any positive real number ε into an approximation set $S(\varepsilon)$ for this ε . By a *quality function* of an approximation scheme S we mean a function $q_S(\varepsilon)$ that transforms ε into the number of bits required for $S(\varepsilon)$.

Definition 4. We say that a function $f(\varepsilon)$ is *asymptotically better* than the function $g(\varepsilon)$ if for every $C > 0$, there exists an $\varepsilon_0 > 0$ such that for $\varepsilon < \varepsilon_0$, $f(\varepsilon) \leq Cg(\varepsilon)$. In other words, f is asymptotically better than g if $f(\varepsilon)/g(\varepsilon) \rightarrow 0$ when $\varepsilon \rightarrow 0$.

Definition 5. We say that an approximation scheme S is *asymptotically better* than an approximation scheme T if the quality $q_S(\varepsilon)$ of scheme S is asymptotically better than the quality $q_T(\varepsilon)$ of scheme T .

Motivation. In principle, we can compare the number of bits $q_S(\varepsilon)$ and $q_T(\varepsilon)$ directly. But even if $q_S(\varepsilon) < q_T(\varepsilon)$ for all ε , we cannot be sure that S is better than T . The reason is that in actual signal processing systems, memory consists of several different unequal parts. For some of them (e.g., for disk memory) it is easy to add additional bits, for some other (e.g., for operational memory) it is more difficult. So, if a scheme S requires a memory that is more difficult to enlarge, then from the user's viewpoint, it can be worse than a scheme T that requires more bits, but those bits are in a memory that is easier to enlarge. In such cases, we must compare not bits themselves, but the total cost of these bits. So, if S requires the memory with a cost c_S per bit, and T requires a memory with a cost c_T per bit, then S is better if $c_S q_S(\varepsilon) \leq c_T q_T(\varepsilon)$. If we denote $C = c_T/c_S$, we can conclude that $q_S(\varepsilon) \leq C q_T(\varepsilon)$.

We would like to say that S is better than T if S is better regardless of the costs, i.e., if the above inequality is true for all $C > 0$. However, if we demand it to be true for all ε , then from $q_S(\varepsilon) \leq C q_T(\varepsilon)$, we conclude that $q_S(\varepsilon) = 0$. So, we can require this inequality only for sufficiently small ε .

Definition 6. We say that two functions $f(\varepsilon)$ and $g(\varepsilon)$ are *asymptotically equivalent* if there exist C_1, C_2 and $\varepsilon_0 > 0$, such that for all $\varepsilon < \varepsilon_0$, we

have $C_1g(\varepsilon) \leq f(\varepsilon) \leq C_2g(\varepsilon)$.

Definition 7. We say that an approximation scheme S is asymptotically of the same quality as an approximation scheme T if their quality functions $q_T(\varepsilon)$ and $q_S(\varepsilon)$ are asymptotically equivalent.

Motivation. This means that for some combination of the costs c_S and c_T , S is better than T , and for some other combination of costs, T is better than S .

Remark. The smallest possible number of elements in a ε -net for F is usually denoted by $N_\varepsilon^F(F)$ (see, e.g., [16], Chapter 10), and the logarithm of this number is denoted by $H_\varepsilon^F(F)$ and is called *metric entropy* of the set F . So, the quality of the best possible approximation scheme is asymptotically equivalent to $H_\varepsilon^F(F_p(\Delta))$.

The idea of using entropy to describe the complexity of neural networks was used in [23] and [2].

III. WAVELET NEURAL NETWORKS AS APPROXIMATORS

A. What Is a Wavelet

It is well known that an arbitrary function on $(0,1)$ can be represented as a Fourier series or a Fourier integral. In other words, an arbitrary function can be represented as a linear combination of waves (sines $\sin(kx)$ and cosines $\cos(kx)$). These component waves are “infinitely long” in the sense that they are different from 0 for arbitrary large x . Since we are interested in approximating functions, that are different from 0 only on the interval $(0,1)$ (or even have a compact support), it seems reasonable to use some kind of “short waves” for approximations, i.e., wave-like functions that tend to 0 as $x \rightarrow \infty$ (or, correspondingly, have compact support). Such functions are called *wavelets*, and they are successfully applied in many areas (see, e.g., [5], [4], [8]).

Depending on our requirements, there exist many different kinds of wavelets. Since we are approximating a function that is located on an interval $(0,1)$, it is reasonable to consider only wavelets with compact support. Since we are interested in approximating p times differentiable functions, it is reasonable to consider p times differentiable wavelets. Let us recall relevant definitions (see, e.g., [8]):

Definition. For an integer m , by a compactly supported 1-dimensional wavelet of class m (or wavelet for short) we mean a function $s(x)$ such that:

- 1) The function s has bounded derivatives up to order m , defined almost everywhere.
- 2) The function $s(x)$ is equal to 0 outside some interval (a, b) .
- 3) $\int x^\alpha s(x) dx = 0$ for $\alpha = 0, 1, 2, \dots, m$.
- 4) The family $\{e_{jk}(x) = 2^{j/2} s(2^j x - k)\}$, where j and

k are arbitrary integers, is an orthonormal basis for $L^2(R)$.

Remarks.

1) The existence of such wavelets was proved in [7], for $a = -Cm$ and $b = Cm$.

2) In the following text, a and b stand for the endpoints of the interval on which $s(x)$ is located.

B. Wavelet Neural Networks

Due to condition (4), an arbitrary smooth function $f(x)$ on $(0,1)$ can be represented by a series $f(x) = \sum_{j,k} C_{jk} 2^{j/2} s(2^j x - k)$ (with $C_{jk} = \int f(x) 2^{j/2} s(2^j x - k) dx$). If we retain finitely many coefficients C_{jk} , we get an approximation for $f(x)$. This approximation is a particular case of the general expression of a 3-layer neural network $\sum_{h=1}^H \beta_h s(w_h x + b_h)$: namely, weights w_h are equal to 2^j , thresholds to $b_h = -k$, and weights β_h on the way from hidden to output neurons are equal to $C_{jk} 2^{j/2}$. In this case, thresholds b_h and weights w_h are “frozen” (i.e., they do not depend on the function $f(x)$).

Let us call such neural networks, for which $s(x)$ is a wavelet, $w_h = 2^j$, and $b_h = -k$, *wavelet neural networks*.

Remark. The relationship between wavelets and neural networks has been explored in [18], [24], [19], [25], [20], [21].

C. Wavelet Neural Network as an Approximation Scheme

We have already mentioned that an arbitrary smooth function $f(x)$ on $(0,1)$ can be represented by a series $f(x) = \sum_{j,k} C_{jk} e_{jk}(x)$, with $C_{jk} = \int f(x) e_{jk}(x) dx$. We can use the coefficients C_{jk} as a record from which we can reconstruct $f(x)$. From the properties of wavelets, we can extract the following definition:

Definition 8. Suppose that Δ and p are given. Let us denote

$$\begin{aligned} C_1 &= \int |y|^p |s(y)| dy, \\ N_2 &= \lceil \max(1/(2p+1) \log_2(18C_1^2(b-a)\Delta^2) - \\ &\quad 2/(2p+1) |\log_2(\varepsilon)|, \\ &\quad 1/(2p) \log_2(18C_1^2\Delta^2) - 1/p |\log_2(\varepsilon)|) \rceil, \\ C_2 &= \max(|s(x)|), \\ N_1 &= \lceil \log_2(9(b-a+1)C_2^2\Delta^2) - 2 \log_2((p+1)!) - \\ &\quad 2 \log_2(\varepsilon) \rceil, \\ \tilde{N} &= (N_1 + 1)(b-a+1) + N_2(b-a) + 2^{N_2+1} - 2, \\ &\quad \text{and} \\ d &= \lceil 1/2 \log_2(9\tilde{N}) - \log_2(\varepsilon) \rceil. \end{aligned}$$

By W we will denote a function that assigns to every

$\varepsilon > 0$ the set $W(\varepsilon)$ of all functions of the type

$$\sum_{j=-N_1}^{N_2} \sum_{k \in (-b, -a+2^j)} \tilde{C}_{jk} e_{jk}(x),$$

where each coefficient \tilde{C}_{jk} is a binary number of the type $p/2^d$, p is an integer such that $p \geq m$, $|\tilde{C}_{jk}| \leq 2^{-j(p+1/2)} C_1 \Delta$ for $j > 0$ and

$$|\tilde{C}_{jk}| \leq 2^{j/2} C_2 \Delta / (p+1)!$$

for $j \leq 0$.

PROPOSITION. *The above-described function W is an approximation scheme for $F_p(\Delta)$.*

Definition 9. *The approximation scheme W from Definition 8 will be called a wavelet neural network approximation scheme, or simply wavelet approximation scheme.*

IV. MAIN RESULT

THEOREM. *For $F_p(\Delta)$, the quality $q_W(\varepsilon)$ of the wavelet approximation scheme W is asymptotically equivalent to $\varepsilon^{-1/p}$.*

The proof is given in [15]. It is based on the estimates provided in [17].

Remark. It is known ([14], [22]) that metric entropy of $F_p(\Delta)$ is asymptotically equivalent to $\varepsilon^{-1/p}$. This means that for every approximation scheme, its quality cannot become better than $\varepsilon^{-1/p}$. Therefore, our Theorem proves that *wavelet neural networks are optimal approximators*, because they require the smallest number of bits for the same precision ε .

Our Theorem is in good agreement with the following result from [9]. Namely, for any basis $e_k(x)$, an arbitrary function can be represented as $\sum_k C_k e_k(x)$. So, instead of the original signal, we can store the coefficients C_k of this expansion. The authors of [9] prove that for a given precision ε , wavelet basis requires the smallest number of coefficients to store. Our Theorem proves that wavelets still lead to the best approximation if we compare them to an arbitrary approximation scheme (not necessarily related to any basis).

REFERENCES

- [1] E. K. Blum, and L. K. Li. "Approximation Theory and feedforward networks", *Neural Networks*, Vol. 4, pp. 511–515, 1991.
- [2] R. W. Brause, "The error-bounded descriptonal complexity of approximation networks", *Neural Networks*, Vol. 6, pp. 177–187, 1993.
- [3] N. E. Cotter, "The Stone-Weierstrass theorem and its application to neural networks", *IEEE Transactions on Neural Networks*, Vol. 1, pp. 290–295, 1990.

- [4] C. K. Chui, **An introduction to wavelets.** and **Wavelets: a tutorial in theory and applications**, Academic Press, 1992.
- [5] J. M. Combes, A. Grossman, and Ph. Tchamitchian (eds.). **Wavelets. Time-frequency methods and phase space**, Springer-Verlag, 1990.
- [6] G. Cybenko. "Approximation by superpositions of a sigmoidal function", *Mathematics of Control, Signals and System*, Vol. 2, pp. 303–314, 1989.
- [7] I. Daubechies, "Orthonormal bases of compactly supported wavelets." *Communications in Pure and Applied Mathematics*, Vol. 61, pp. 909–996, 1988.
- [8] I. Daubechies, **Ten lectures on wavelets**, SIAM, 1992.
- [9] R. A. DeVore, B. Jawerth, and B. J. Lucier. "Image compression through wavelet transform coding", *IEEE Transactions on Information Theory*, Vol. 38, pp. 719–746, 1992.
- [10] K. Funahashi, "On the approximate realization of continuous mappings by neural networks", *Neural Networks*, Vol. 2, pp. 183–192, 1989.
- [11] R. Hecht-Nielsen, "Kolmogorov's mapping neural network existence theorem", *IEEE International Conference on Neural Networks*, SOS Printing, San Diego, 1987, pp. 11–14.
- [12] K. Hornik, "Approximation capabilities of multilayer feedforward networks", *Neural Networks*, Vol. 4, pp. 251–257, 1991.
- [13] K. Hornik, M. Stinchcombe, and H. White. "Multilayer feedforward networks are universal approximators", *Neural Networks*, Vol. 2, pp. 359–366, 1989.
- [14] A. N. Kolmogorov and V. M. Tikhomirov, "The ε -entropy and ε -capacity of sets in function spaces", *Uspekhi Mat. Nauk*, Vol. 14, No. 2 (86), pp. 3–86, 1959; English transl. in *Amer. Math. Soc. Transl.*, Series 2, Vol. 17, pp. 277–364, 1961.
- [15] V. Kreinovich, O. Sirisaengtaksin, S. Cabrera, *Wavelet neural networks are optimal approximators for functions of one variable*, University of Texas at El Paso, Computer Science Department, Technical Report No. UTEP-CS-92-29, 1992.
- [16] G. G. Lorentz, **Approximation of Functions**, Holt, Rinehart and Winston, 1966.
- [17] Y. Meyer, **Wavelets and Operators**, Cambridge University Press, 1992.
- [18] Y. C. Pati and P. S. Krishnaprasad, "Discrete affine wavelet transforms for analysis and synthesis of feedforward neural networks", In: R. Lippman, J. Moody, and D. Touretzky (eds.),

- Advances in Neural Information Processing Systems. 3**, pp. 743–749, Morgan Kaufmann, 1991.
- [19] Y. C. Pati and P. S. Krishnaprasad, “Analysis and synthesis of feedforward neural networks using discrete affine wavelet transforms”, *IEEE Transactions on Neural Networks*, 1992.
- [20] H. Szu, B. Telfer, and S. Kadambe, “Neural Network Adaptive Wavelets for Signal Representation and Classification,” *Opt. Eng.*, Vol. 31, pp. 1907–1916, 1992.
- [21] H. Szu, X.-Y. Yang, B. Telfer, and Y. Sheng, “Neural Network and Wavelet Transform for Scale-Invariant Data Classification,” *Phys. Rev. E*, Vol. 48, pp. 1497–1501, 1993.
- [22] A. G. Vitushkin, **Theory of the transmission and processing of information**, Pergamon Press, 1961.
- [23] R. C. Williamson, “ ϵ -entropy and the complexity of feedforward neural networks”, In: R. Lippman, J. Moody, and D. Touretzky (eds.), **Advances in Neural Information Processing Systems. 3**, pp. 946–952, Morgan Kaufmann, 1991.
- [24] Q. Zhang and A. Benveniste, “Approximation by nonlinear wavelet networks”, *Proc. 1991 IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, Toronto, Ontario, Canada, May 14–17, 1991 (IEEE Press).
- [25] Q. Zhang and A. Benveniste, “Wavelet networks”, *IEEE Trans. on Neural Networks*, Vol. 3, pp. 889–898, 1992.