Using the Wavenet for function approximation

Alexander Ypma and Robert P. W. Duin

Pattern Recognition Group, Faculty of Applied Physics, Delft University of Technology
Lorentzweg 1, 2628 CJ, Delft,The Netherlands
email: ypma@ph.tn.tudelft.nl

Keywords: neural networks, wavelets, function approximation, machinery monitoring

Abstract

When the aim is to make an arbitrary nonlinear mapping, neural networks are known to be a suitable technique. The Wavenet combines them with the wavelet transform, enabling a multi-scale approximation, while dilation and translation parameters can be fit to the data. Some properties of the Wavenet are investigated and an outlook to application in machinery monitoring is provided.

1 Introduction

Neural networks can be approached in several ways. Multilayered Perceptrons are capable of implementing any nonlinear decision surface with respect to a given set of data [12], which is valuable for pattern recognition purposes. Alternatively, they can be looked upon as a vehicle for making arbitrary continuous mappings from some input space to an output space [6], which is useful for system modeling and identification [1].

In this paper we focus on the use of feedforward networks for function approximation, employing a gradient descent training procedure. Recently, Znang, êt al.[1] introduced the wavelet network (or *Wavenet*) for this purpose. Based on wavelet theory, they developed a novel network architecture, capable of establishing (in principle) any nonlinear mapping. Guaranteed upper bounds on the accuracy of approximation by the Wavenet were established in [3].

In this paper we investigate some properties of the Wavenet for function approximation. First, we will take a look at a possible application of the Wavenet for predictive maintenance of rotating machinery.

2 Nonstationary analysis in machinery monitoring

The purpose of machine condition monitoring is to be able to detect and diagnose machine wear in order to perform predictive maintenance, usually on the basis of a machine's vibration behaviour. A representative spectrum can be used to characterize a machine uniquely (hence act like a machine *fingerprint*), and the conventional approach is to diagnose machine health by looking at deviations of the current spectrum with the fingerprint [7]. Diagnosis can be done using neural networks, where e.g. vibration spectra are preprocessed (compressed), and classified by a neural network into classes of machine pathologies like unbalance, misalignment, bearing failure and absence of machine faults [9].

Modern approaches focus on the time-frequency characteristics of certain machine faults: incipient failures are often not repetitive in the earliest stages (e.g. an occasional clash occurs), and give rise to nonstationary, impulsive (low-energy broadband) signals, often occuring over several scales. It is believed that the path toward progress in the area is simultaneous time-frequency analysis of machine vibration signals [8]. Hence, a future application would be to use a Wavenet for the extraction of multi-scale features from machine vibration signals, probably enabling better compression then linear compression methods. Using these features, a generic description of a machine's normal behaviour under different cirmcumstances could be made (the *generalized fingerprint*, figure 1), serving as a reference for the detection of fault states (i.e. *novelty*).
In order to be able to obtain localized information about a certain signal or function in both time and frequency domain, one could start with windowing the Fourier transform, leading to the *short-time Fourier transform* [11]

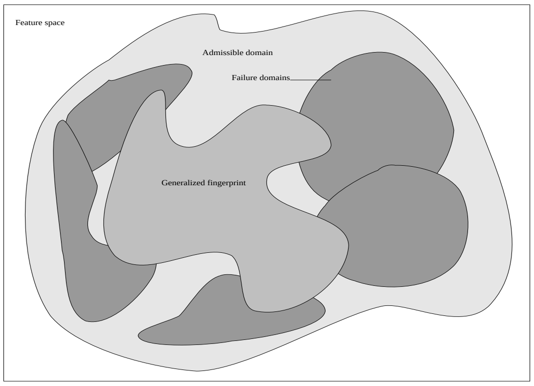

Figure 1: **Generalized fingerprint**

$$(S_w f)(b,\omega) = \int_{-\infty}^{\infty} e^{-i\omega t} f(t)\overline{w(t-b)}dt \qquad (1)$$

where $w \in L^2(R)$ is a window function (e.g. a Gaussian), and b, ω are translation and modulation parameters, respectively. This leads to a time-frequency window (i.e. the part of time and frequency axis, where function w and its Fourier transform $\hat{w}$ achieve their most significant values) with boundaries $[\mu_w + b - \sigma_w, \mu_w + b + \sigma_w]$ and $[\mu_{\hat{w}} + \omega - \sigma_{\hat{w}}, \mu_{\hat{w}} + \omega + \sigma_{\hat{w}}]$, where $\mu_w, \mu_{\hat{w}}$ and $2\sigma_w, 2\sigma_{\hat{w}}$ are the center and width of the time-frequency window in both domains. Note that width $2\sigma_w$ and height $2\sigma_{\hat{w}}$ of the windows are constant, independent of the center of the window, which is undesirable when the objective is to obtain spectral information about limited time intervals in various frequency bands.

In order to achieve accurate high-frequency information, the time interval should be short, and vice versa. This can be established by introducing the *Continuous Wavelet Transform*

$$(W_\psi f)(b,a) = \int_{-\infty}^{\infty} f(t)\overline{\psi_{a,b}(t)}dt, a \in R^+, b \in R \qquad (2)$$

where $\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\psi(\frac{t-b}{a})$ is a set of functions, called *wavelets*, obtained from a basic wavelet $\psi \in L^2(R)$ by dilation (i.e. scaling) and translation. The basic wavelet has to satisfy the admissibility condition

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty \qquad (3)$$

which in fact states that ψ must have zero mean (and hence has a "wave character"). When ψ and its Fourier transform $\hat{\psi}$ are both window functions, the continuous wavelet transform (2) of a signal f gives localized information in $[b + a\mu_\psi - a\sigma_\psi, b + a\mu_\psi + a\sigma_\psi]$ and $[\frac{\mu_{\hat{\psi}}}{a} - \frac{1}{a}\sigma_{\hat{\psi}}, \frac{\mu_{\hat{\psi}}}{a} + \frac{1}{a}\sigma_{\hat{\psi}}]$. Now, the width of the frequency window decreases for increasing scale parameter a, and vice versa.

Note that since the ratio of center-frequency to frequency bandwidth is constant, the wavelet decomposition gives rise to a *constant percentage bandwidth analysis*, which is a much practiced technique in the field of machinery vibration analysis [10].

3 Wavelet theory

A wavelet transformed function can be recovered from its wavelet components by the inversion formula [11]

$$f(x) = \frac{1}{C_\psi}\int_{\infty}^{\infty}\int_{\infty}^{\infty}(W_\psi f)(b,a)\psi_{a,b}(x)\frac{dadb}{a^2} \qquad (4)$$

In practice, f has to be recovered from a finite number of wavelet components, so f has to be reconstructed from samples of the continuous wavelet transform. This means that in (4) only a discrete number of wavelets can be used for reconstruction. When these wavelets constitute an *orthonormal* basis of the $L^2(R)$, perfect reconstruction is still possible. When the wavelets $\psi_{b_0;j,k} = 2^{-j/2}\psi(2^{-j}x - kb_0)$ form a *frame*, i.e.

$$A\|f\|_2^2 \leq \sum_{j,k\in Z} |< f, \psi_{b_0;j,k} >|^2 \leq B\|f\|_2^2 \qquad (5)$$

for some constants $A, B \in R_+$, and $b_0 \in R_+$, *approximate* reconstruction can be done by

$$f_{app} = \frac{2}{A+B}\sum_{j,k\in Z} < f, \psi_{b_0;j,k} > \psi_{b_0;j,k} \qquad (6)$$

where $< \cdot, \cdot >$ stands for the inner product in $L^2(R)$. The approximation becomes better when B/A is closer to one.

The theory is now extended to functions $f : R^n \mapsto R$, for which a function $\psi : R^n \mapsto R$ is to be found such that the denumerable family of n-dimensional wavelets [1]

$$\Phi = \{|D_k^{\frac{1}{2}}|\psi[D_k\mathbf{x} - \mathbf{t}_k]\} \qquad (7)$$

with $\mathbf{t}_k \in R^n, D_k = diag(\mathbf{d}_k), \mathbf{d}_k \in R_+^n, k \in Z$ forms a frame. Here, $|.|$ stands for the determinant of a matrix, boldface symbols indicate n-dimensional variables.

Note that for the multi-dimensional case the matrix D_k plays the role of the scaling parameter a of the scalar case: it consists of $d_{ij} = \frac{1}{a_{ij}}, i = 1, \ldots n; j = 1 \ldots, N$, where N is the number of wavelets used for the approximation.

As a consequence of the frame property, the family Φ is *dense* in $L^2(R^n)$. Then the approximation

$$g(\mathbf{x}) \approx \sum_{i=1}^{N} w_i \psi[D_i \mathbf{x} - \mathbf{t_i}] + \bar{g} \qquad (8)$$

can be used to reconstruct f.

Note: it can be shown that the frame property is satisfied when we take as ψ the function $\prod_{i=1}^{n} x_i e^{-\frac{1}{2}x_i^2}$, the direct product of one-dimensional Gaussian derivatives.

4 Wavelet networks

Reconstruction formula (8) can be emulated with a multilayered Perceptron consisting of one hidden layer of "wavelons", i.e. units i with activation $D_i(\mathbf{x} - \mathbf{t_i})$ and transfer function ψ, and one output with inner product activation and linear transfer function. A bias $\bar{g}$ is included to deal with nonzero mean functions on finite domains.

Just as with radial-basis function networks, where weights and kernel positions and spreads can be adapted with a supervised learning procedure [15], the free parameters of the Wavenet (denoted by θ, i.e. output bias, weights, and wavelet positions and supports) can be tuned to the function to be approximated by a least-mean-squares algorithm.

Consider a set of learn samples $\{\mathbf{x}_k, y_k = f(\mathbf{x}_k) + v_k\}$, where $\{v_k\}$ is observation noise. Define by $g_\theta(\mathbf{x})$ the output of the network parameterized by θ. Then the parameters have to be selected such that the mean-squared error $C(\theta)$ between targets and network predictions over a batch of training samples is minimized. This can be done iteratively by gradient descent, i.e. computing the gradient of the function to be minimized w.r.t. all the parameters of the model, and updating by [15]

$$\theta(n+1) = \theta(n) - \gamma \nabla \theta \qquad (9)$$

Here, γ is a parameter controlling the learning speed. When updating is done after each training sample $(\mathbf{x}_k, y_k)$, a *stochastic* gradient descent procedure is obtained, which can be helpful to avoid local minima in the error surface [15].

Taking the nature of the basis functions into account, some heuristics constraining the adjustable parameters and guiding network initialization can be formulated:

initialization: since the a_i and t_i denote scale and center of the wavelets that are used to approximate f, we can exploit this by positioning the wavelets initially in a sensible manner, probably speeding up the convergence process. Split up the domain of approximation D dyadically (i.e. the first wavelon acts on the whole domain D, the second and third on the left and right halves of the original domain, and repeat until all N wavelons have been initialized), and choose a_i, t_i accordingly. The output bias can be chosen initially as the mean of the function to be approximated, whereas the weights can be initially set to zero.

parameter constraints: wavelets that are drifting outside the domain of approximation, or having support near zero will not aid much in the overall approximation. Hence, truncate, i.e. fix the parameters at some boundary value, a_i, t_i during learning using some prespecified parameters E, ε such that:

- $t_i \in E, i = 1, \ldots, N; D \subset E \subset R^n$
- $D_i^{-1} > \varepsilon I, i = 1, \ldots, N$

The specific choices for the parameters E, ε can influence the accuracy of the resulting approximation, and any heuristics about how to choose them are absent.

5 Experiments

The parameters that have to be prespecified in experiments with wavelet networks are: number of wavelons N, number of learning epochs, learning rate γ, and projection parameters ε, E, assuming the size n of the train set is fixed. In [3] the number of wavelons is estimated as $N \approx \sqrt{n}$, which enables us to do an educated guess about the domain in which to experiment. Also, the accuracy of approximation is said to depend on the dimension d of the function f to be approximated and its complexity ρ: Wavenets can achieve an L^2-error of order $N^{-\frac{\rho}{d}}$, where ρ denotes the regularity, i.e. the number of summable derivatives, of f [3]. Moreover, the number of training epochs used in [1] and [2] were of the order 10^4, the learning rate was set at 0.1.

We performed some preliminary experiments with the Wavenet, using two target functions of varying complexity: a simple parabola on the domain $[-10, 10]$, and a curved surface $f(x_1, x_2) = (x_1^2 - x_2^2) sin(0.5 x_1)$ on the domain $[-10, 10] X [-10, 10]$. For ψ we chose the function $\prod_{i=1}^{n} x_i e^{-\frac{1}{2}x_i^2}$, where $n = 1$ for the first function, and $n = 2$ for the second.

The parabola was learned (using pattern mode updating) with 50 training samples, hence some 7 wavelons should be adequate for a good approximation (the parabola is a very smooth function

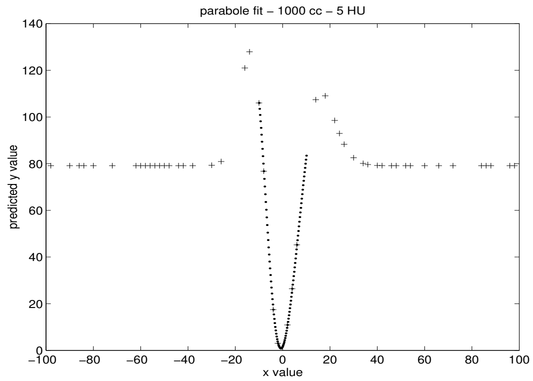

Figure 2: **Wavenet approximation 5 wavelons**

of dimension one). We tried Wavenets of 5, 15, 25 and 35 wavelons for this task (default: 35). We trained the network during 1000 epochs. In all cases, the approximation was very good (fig.2, dotted line), sometimes already after very few updates. When the wavnet was tested over a larger domain, the +-line was observed (fig.2).

Increasing the number of wavelons leads to similar train errors. When the Wavenet with 35 wavelons is tested outside its training domain, it can be seen that there are in fact too many degrees of freedom for fitting the simple parabolic function (fig.3). For more complex functions, this will probably lead to overfitting of the data.

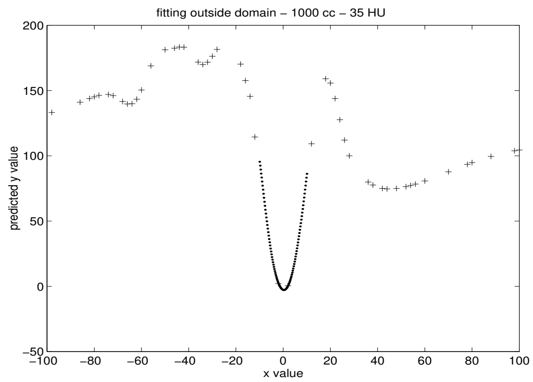

Figure 3: **Wavenet approximation 35 wavelons**

Suitable network initialization was observed to speed up the approximation process: in one experiment, the wavelets were all positioned correctly, and training required only one epoch.

We chose the projection parameters as:

- $E_{lower} = D_{lower} - C_1(D_{upper} - D_{lower})$
- $E_{upper} = D_{upper} + C_1(D_{upper} - D_{lower})$
- $\varepsilon = \frac{0.5(D_{upper} - D_{lower})}{NC_2}$

and varied the values of C_1, C_2 as $\{0, 2, 5, 10\}$ and $\{1, 2, 3, 5\}$, respectively (default: $C_1 = 5, C_2 = 2$). Also, in one set of experiments we left out the truncation completely. No significant differences w.r.t. learning behaviour and final approximation could be found, i.e. in all cases the parabola was fitted fairly easy by the network.

Finally, also the learning rate γ was varied as $\{0.1, 0.3, 0.5, 0.9\}$ (default: 0.5). For all choices of γ, the fit was very good, except for $\gamma = 0.9$, where just a straight line was learned. This indicates that in this case the steps taken in the parameter space were too large: overshoot of the error minimum requires longer training time.

For the second function, approximation results are still not satisfactory. We trained on a set of 110 samples, for 20.000 epochs with learning rate 0.3, using 25, 50 and 75 wavelons, and varied the remaining parameters in the same way as with the first function. A typical learning curve is shown in fig.4.

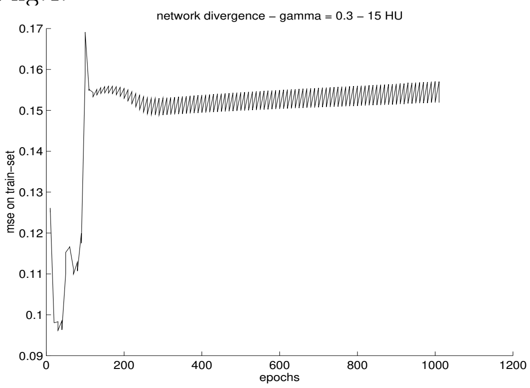

Figure 4: **Learning curve for curved surface**

Oscillatory behaviour during the learning process is visible, while also a local minimum seems to be overshooted. Inclusion of a small *momentum term* (0.01) proved be valuable in later experiments. Moreover, setting the learning rate to a very small value (0.05) seemed to improve convergence. We trained a Wavenet of 150 wavelons for 5000 cycles, which showed a gradual improvement in train error (to approx. 0.05 after some 600 cycles), followed by a steady oscillation around this value, probably indicating a local minimum.

6 Discussion

In [3] it was already noted that the surprising effectiveness of the experiments in [1] was mainly due to their initialization procedure, taking the wavelet decomposition into account. For the simple parabola function, the benefit of adequate initialization was indeed observed. The amount of parameters was limited in this case, and smoothness of the function to be approximated enabled accurate approximation.

The same authors mention that training procedures may be notoriously slow: since the approximation class is nonlinear in the adjustable parameters, the gradient descent optimization criteria are highly nonconvex functions, exhibiting many local minima. This is in agreement with our findings for the approximation of the second function. Probably due to the huge amount of parameters that have to be adapted in this case, the error surface exhibits many local minima, leading to poor convergence.

Also, a higher-dimensional input space brings about a "curse of dimensionality" problem: increasing the the input dimension would require an exponential increase in the number of training patterns covering the whole input space. The rate of convergence will improve with lower dimensional inputs [5], which is in accordance with our experiments.

A possible remedy could be to include a momentum term into the update formula (9), in order to increase the rate of learning yet avoiding the danger of instability or ending up in a local minimum on the error surface [15], which indeed seemed to be helpful in some later experiments.

As an alternative, incorporation of a different initialization procedure (backward elimination selection of candidate wavelets, [4]) was proposed as a good alternative to the somewhat heuristic initialization procedure employed in [1] and this paper.

In [5], an algorithm was proposed where truncation of wavelets amounted to putting them to zero, as opposed to our fixation step. Using this algorithm, the parameters of the Wavenet could be obtained by a direct Monte-Carlo procedure [3], which avoids the need for a stochastic gradient descent procedure, sensitive to undesirable local minima, altogether.

7 Acknowledgment

This work was partially supported by the Foundation for Applied Sciences (STW) of the Netherlands, project no. DTN-44.3584.

References

[1] Q. Zhang, A. Benveniste, *Wavelet Networks*, IEEE Trans. Neural Networks, Vol. 3, No. 6, Nov. 1992, pp. 889 - 898

[2] R. Baron, H. Paugam-Moisy, *Using PVM for the selection of Wavelet Neural Network parameters*, Research report No. 94-41, Ecole Normale Supérieure de Lyon, Lyon FR, 1994

[3] B. Delyon, A. Juditsky, A. Benveniste, *Accuracy analysis for wavelet approximations*, IEEE Trans. Neural Networks, Vol. 6, No. 2, March 1996, pp. 332 - 347

[4] Q. Zhang, *Regressor selection and wavelet network construction*, Technical Report No. 709, IRISA/ INRIA, April 1993

[5] A. Juditsky, H. Hjalmarsson, A. Benveniste, B. Delyon, L. Ljung, J. Sjöberg, Q. Zhang, *Nonlinear black-box models in system identification: mathematical foundations*, Technical Report, IRISA/ INRIA, June 27, 1995

[6] K. Funahasi, *On the approximate realization of continuous mappings by neural networks*, Neural networks, Vol. 2, 1989, pp. 183 - 192

[7] J. S. Mitchell, *An introduction to machinery analysis and monitoring*, PennWell Publ. Comp., 1981

[8] M. J. Dowling, *Application of non-stationary analysis to machinery monitoring*, IEEE, 1993

[9] G. A. P. Fontaine, E. E. E. Frietman, R. P. W. Duin, *Preventive and predictive maintenance using neural networks*, J. Microelectric systems integration, Vol. 4, No. 2, 1996

[10] R. B. Randall, *Frequency analysis*, Brüel & Kjaer, DE, 1987

[11] J. B. T. M. Roerdink, *Wavelets for signal and image processing*, Lecture notes, Dept. of Computing Science, Rijksuniversiteit Groningen, NL, 1993

[12] W. F. Schmidt, *Neural pattern classifying systems*, PhD-thesis, Delft University of Technology, Delft NL, 1994

[14] B. D. Ripley, *Pattern recognition and neural networks*, Cambridge University Press, Cambridge, NY, USA, 1996

[15] S. Haykin, *Neural networks, a comprehensive foundation*, Macmillan College Publishing Company, Inc., New York, NY, USA, 1994