Application of neural networks to the inverse light scattering problem for spheres

Zbigniew Ulanowski, Zhenni Wang, Paul H. Kaye and Ian K. Ludlow

A new approach suitable for solving inverse problems in multi-angle light scattering is presented. The method takes advantage of multidimensional function approximation capability of radial basis function (RBF) neural networks. An algorithm for training the networks is described in detail. It is shown that the radius and refractive index of homogenous spheres can be recovered accurately and quickly, with maximum relative errors of the order of 10^{-3} and mean errors as low as 10^{-5}. The influence of the angular range of available scattering data on the loss of information and inversion accuracy is investigated and it is shown that more than two thirds of input data can be removed before substantial degradation of accuracy occurs.

Key words: Light scattering, particle sizing, sphere, inverse problem, neural network, radial basis function.

1. Introduction

Over the last hundred or so years great advances have been made in developing the electromagnetic theory of scattering from particulate matter. Accurate predictions of the properties of scattered electromagnetic fields - the direct scattering problem - are now possible in many situations. Rigorous solutions exist for numerous particle types, such as homogeneous and inhomogeneous spheres, ellipsoids, cylinders, generalised axisymmetric particles and others. Of far greater practical importance, however, is the determination of properties of particles from the knowledge of scattered fields - the inverse scattering problem. Solving this type of problem is required in numerous applications, ranging from astronomy and remote sensing, through aerosol and emulsion characterisation, to non-destructive analysis of single particles and living cells.[1-4]

Unfortunately, the inverse problem has proven to be much less tractable, even for the simplest particle shapes. This is partly because in many cases inverse problems do not have unique solutions and are therefore ill-posed. Lack of rigorous solutions has motivated the development of methods

The authors are with the Department of Physical Sciences, University of Hertfordshire, Hatfield, Herts AL10 9AB, UK. E-mail: z.ulanowski@herts.ac.uk

based on approximate models of scattering for some particle geometries, for example, assuming that the particles are weak (Rayleigh-Debye) scatterers or that diffraction alone can adequately describe the interaction process. The reader is referred to the large body of existing literature and several monographs for further details.[5-8] However, when such methods are inappropriate, empirical procedures have to be used which are based on generating solutions to the direct problem (after making assumptions concerning the shape, internal structure of the particle, etc.) and matching these solutions to experimental data.[1-3,9,10] These procedures can be very slow, difficult to implement and require substantial computing resources. Attempts to apply numerical optimisation methods to assist the fitting process have been only moderately successful so far. Two major difficulties are apparent. The first one is the presence of numerous local solutions necessitating the use of cumbersome and slow global optimisation methods. The second difficulty arises in cases where scattering data is "noisy" or distorted, for instance when it originates from single, small particles. Under such circumstances spurious solutions appear, leading to large errors in the determination of particle parameters.[11-13]

A natural starting point for attempts to solve the inverse scattering problem for fine particles is the simple case of a single, homogeneous, non-absorbing sphere. If a plane incident wave of known wavelength and state of polarisation and a known medium surrounding the particle are assumed, the scattering from the particle can be completely described using two parameters, namely its radius and refractive index. The direct problem can then be solved using the series expansions of the Lorenz-Mie theory.[5-7] In addition, if we assume that the irradiance of the light scattered by the particle is measured in one plane only, then the scattering can be described by a function of the scattering angle (defined as the angle between the direction of the incident wave and the direction of observation). This arrangement leads to a one-dimensional scattering "pattern" which is representative of the properties of the particle and has been used as a basis for characterisation of both single particles and particle distributions.[1-3,8] The present study is confined to such a scattering geometry. However, extending it to more general cases, with larger numbers of parameters, should not present major difficulties, as long as solutions to the direct problem are available.

One motivation behind the present study has been the observation that simple least squares fitting methods often fail to locate correct solutions for noisy and/or distorted scattering data. Minimising a merit function of the type:

$$\frac{1}{N}\sum_{i=1}^{N}\left[I(\theta_i) - c\, M(\theta_i)\right]^2 \qquad\qquad (1)$$

where $I(\theta_i)$ is a theoretical value for the scattered light irradiance at the scattering angle θ_i, $M(\theta_i)$ is a measured quantity proportional to the scattered irradiance at the angle θ_i and c is an instrumental constant (usually not known accurately), is often an unsatisfactory approach.[2,11,12] Using additional information, such as the positions of the peaks in the light scattering pattern,[2,14] or weighting the data in various ways[1] can lead to substantial improvements, although these approaches are still heuristic to some extent. A promising method, which directly yields particle radius but may eventually lead to a full inverse solution, is based on transforming the patterns in a set of Legendre or Gegenbauer polynomials.[15-17] Particle size distributions as well as complex refractive indices can be obtained under some constraining conditions by combining empirical and analytical (eigenfunction) methods.[8] In the context of the present study it is significant that visual comparison of experimental and theoretical data - taking advantage of intrinsic data selection and processing capabilities of the human brain - often produces good results, even where other approaches fail.[2] These facts indicate that a degree of, possibly quite complex, data pre-processing is required before methods such as least squares fitting or numerical optimisation can be successfully applied. On this basis it can be conjectured that artificial neural networks should be well suited to solving the present problem. Indeed, recent work on the determination of particle size distributions[18] and particle classification[19] using multi-angle light scattering data, as well as earlier studies on the recovery of size distributions from backscattering[20] have shown that neural networks can be used to solve some classes of inverse problems in light scattering. Moreover, the neural network approach may eventually yield several by-products, e.g.: data reduction methods, optimal data collection methods, efficient algorithms for comparing theoretical and experimental data and, last but not least, insights into the theory of scattering.

An approach to solving inverse problems in multi-angle light scattering based on radial basis function (RBF) neural networks is proposed. The method takes advantage of the capability of the RBF networks to approximate multidimensional functions. A detailed investigation of the technique for the case of homogeneous, spherical particles is presented. A step-by-step algorithm for training the networks is described. Testing of the networks is carried out using data generated from the

Lorenz-Mie theory. Several scaling and weighting schemes for the scattering data are investigated. The influence of the angular range of available scattering data on inversion accuracy is also studied.

2. Neural network algorithm

In the context of the present study the inverse scattering problem can be stated as follows. If a homogeneous particle, immersed in a given external medium, is illuminated by light of a given wavelength and state of polarisation, the intensity of scattered light at angle θ with respect to the forward direction is a function of the radius and the refractive index (r, n) of the particle. This function can be given as:

$$I(\theta) = F(\theta, r, n) , \qquad\qquad (2)$$

where the function F is defined by the Lorenz-Mie theory.[5-7] Given a vector of discrete measurements $\mathbf{x}$ of scattered light intensity $I(\theta)$ at m different scattering angles θ, the inverse scattering problem is to determine values of r and n from the measurements $\mathbf{x}$. In other words, it is to find an inverse function $f = F^{-1}$ such that r and n can be determined by $f(\mathbf{x})$. In the artificial neural network approach a set of discrete intensity patterns $I(\theta)$ generated from the Lorenz-Mie theory together with the parameters r and n are used as network training examples in order to form an approximation of the inverse function f. It will be assumed that the patterns are sufficiently dense to avoid spatial aliasing.

A radial basis function (RBF) neural network has simple architecture consisting of only one hidden layer. Fig. 1 shows a generalised architecture of such a network with m inputs, N hidden nodes and k outputs. The hidden nodes are radial basis functions and the network output is simply a linear summation of the weighted basis functions. The radial basis function $\omega(\mathbf{x}, \mathbf{c})$ is a non-linear function solely dependent on a radial distance $\| \mathbf{x} - \mathbf{c} \|$, where $\mathbf{c}$ is the function's "centre". In general, the distance need not be Euclidean, although it often is, as in the case of the present study. The implementation of an RBF network includes selecting the basis function form and parameters and finding the weights.

Suppose the function to be approximated is $f : \mathbf{x} \rightarrow \mathbf{y}$, where $\mathbf{x} \in \mathbf{R}^m$ and $\mathbf{y} \in \mathbf{R}^k$. The RBF network approximation of f , $\overline{f}$, will have the following form:

$$\mathbf{y} = \overline{f}(\mathbf{x}) = \sum_{i=1}^{N} \mathbf{w}_i \, \omega_i(\mathbf{x}, \mathbf{c}_i) + \mathbf{b}, \qquad\qquad (3)$$

where $\mathbf{x} \in \mathbf{R}^m$ is the input vector, $\mathbf{y} \in \mathbf{R}^k$ is the output vector, N is the number of hidden nodes, $\mathbf{w}_i \in \mathbf{R}^k$ is a weight vector, $\omega_i(\mathbf{x}, \mathbf{c}_i)$ are radial basis functions corresponding to the hidden nodes in Fig. 1, $\mathbf{c}_i \in \mathbf{R}^m$ is a centre vector, and $\mathbf{b} \in \mathbf{R}^k$ is a bias vector. The bias vector is used to compensate for the difference between the mean of the output vector and the corresponding target.

A previous study[19] described a simplified RBF network in which the basis functions are made Gaussian and take the form:

$$\omega_i = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{d_i^2} \right), \tag{4}$$

where $\|\mathbf{x} - \mathbf{c}_i\|$ is Euclidean distance and d_i are the widths. This simplification essentially consists in making the receptive fields of the basis functions spherical, while in general they can be ellipsoidal. The reason for the use of this model is to reduce complexity, which is particularly important in the high-dimensional application considered here. Although the simplification could affect network performance, accuracy can be recovered by scaling the input data to compensate for the changed receptive fields.

Given a sufficient number of input and output examples, $\mathbf{x}_j$ and $\mathbf{y}_j$, respectively, the RBF network can be trained by forming the interpolation:

$$\mathbf{y}_j = \sum_{i=1}^{N} \mathbf{w}_i \exp\left(-\frac{\|\mathbf{x}_j - \mathbf{c}_i\|^2}{d_i^2} \right) + \mathbf{b}, \qquad j = 1, 2, ..., S, \tag{5}$$

where S is the number of example input/output pairs. The unknown parameters in this model are the weight vectors $\mathbf{w}_i$, the centre vectors $\mathbf{c}_i$, the widths d_i and the bias vector $\mathbf{b}$. If the centres and the widths are known, training is a matter of solving the following linear equation:

$$[\mathbf{y}_1 \ \mathbf{y}_2 \ ... \ \mathbf{y}_S] = [\mathbf{w}_1 \ \mathbf{w}_2 \ ... \ \mathbf{w}_N]\Phi + \mathbf{b}\mathbf{a}, \tag{6}$$

where $\mathbf{y}_j = [y_{j1} \ y_{j2} \ ... \ y_{jk}]^T$ is the j-th example output vector, $\mathbf{w}_i = [w_{1i} \ w_{2i} \ ... \ w_{ki}]^T$ is the i-th node weight vector, $\mathbf{b} = [b_1 \ \ b_2 \ \ ... \ b_k]^T$ is the bias vector ($[.]^T$ denotes matrix transpose), $\mathbf{a} = [1 \ \ 1 \ \ ... \ \ 1]$ and Φ is a matrix of i-th node radial basis functions calculated for the j-th example input vector:

$$\Phi = \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1S} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2S} \\ \cdots & \cdots & \cdots & \cdots \\ \phi_{N1} & \phi_{N2} & \cdots & \phi_{NS} \end{bmatrix}$$

$$\phi_{ij} = \exp\left(-\frac{\|\mathbf{x}_j - \mathbf{c}_i\|^2}{d_i^2} \right)$$

If the bias vector is included in the weight matrix, then Eq. (6) can be written in the following compact form:

$$\mathbf{Y} = \mathbf{W_b}\,\Phi_{\mathbf{a}}, \qquad\qquad (7)$$

where

$$\mathbf{Y} = [\,\mathbf{y}_1 \;\; \mathbf{y}_2 \;\; \cdots \;\; \mathbf{y}_S\,],$$
$$\mathbf{W_b} = [\,\mathbf{w}_1 \;\; \mathbf{w}_2 \;\; \cdots \;\; \mathbf{w}_N \;\; \mathbf{b}\,],$$
$$\Phi_{\mathbf{a}} = \begin{bmatrix} \Phi \\ \mathbf{a} \end{bmatrix}.$$

The network is trained by obtaining the least squares solution of Eq. (7). Consequently, the weights

$$\mathbf{W_b} = \mathbf{Y}\Phi_{\mathbf{a}}^{\mathrm{T}}(\Phi_{\mathbf{a}}\Phi_{\mathbf{a}}^{\mathrm{T}})^{-1}. \qquad\qquad (8)$$

are given by

Before the weights are computed, the centres and the widths must be chosen. To maximise accuracy, each training input is made to be a centre, and, as a result, the number of hidden nodes in the RBF network is equal to the number of training input vectors, i.e. $N = S$. Since the widths d_i control the degree of overlapping of the m-dimensional Gaussian functions, they control the generalisation ability of the RBF network. In order to obtain good generalisation, the choice of the width d_i should ensure that the i-th basis function responds to inputs furthest from its centre $\mathbf{c}_i$. However, this requirement conflicts with the need to approximate functions locally. To satisfy these opposing demands, d_i can

be calculated as the difference between maximum and minimum Euclidean distances from the centre vector $\mathbf{c}_i$ to all other training inputs, namely:

$$d_i = d\left(\max_{j \neq i} \left\| \mathbf{c}_i - \mathbf{c}_j \right\| - \min_{j \neq i} \left\| \mathbf{c}_i - \mathbf{c}_j \right\| \right), \qquad j = 1, 2, \ldots, S. \qquad (9)$$

where d is a constant. As a result, the wider the data spread, the larger the width and the network is automatically adjusted to different input data. The constant d has, typically, the value of one but it can be altered to provide "fine tuning" of the network.

Within the framework provided by Eq. (5), the inverse scattering problem can be solved by unsupervised training (using only the inputs) consisting of selecting the centres and calculating the widths from Eq. (9), followed by supervised training (using the inputs and the outputs) to determine the weights from Eq. (8).

3. Neural network testing

Simulations were conducted using theoretical data generated from the Lorenz-Mie theory. A wavelength of 0.5145µm, incident light polarised perpendicularly to the scattering plane and a refractive index of the surrounding medium (water) of 1.336 were assumed. Discrete ranges of parameter values, $n \in [1.5, 1.7]$ (dimensionless), and $r \in [0.5, 1.5]$ micrometers, were used to construct a matrix of parameter pairs, each with a corresponding normalised intensity pattern $x(\theta)$. For logarithmically scaled data the patterns were calculated from:

$$x(\theta) = 1 - \log|S_1(0)|^2 + \log|S_1(\theta)|^2, \qquad (10)$$

where $S_1(\theta)$ is the element of the scattering matrix corresponding to perpendicular polarisation.[5-7] Note that, since $|S_1(\theta)|^2 \leq |S_1(0)|^2$, $x(\theta) \leq 1$ and $x(0)=1$. The motivation behind the use of this scaling was that, generally, measurement accuracy of scattering functions such as $S_1(\theta)$ is much better in terms of relative values (for different scattering angles) as opposed to absolute ones - the measurements are "differential". For linear data the patterns were calculated from:

$$x(\theta) = g(\theta) |S_1(\theta)|^2, \qquad (11)$$

where $g(\theta)$ is an appropriate weighting function. The intensity patterns, sampled between 0° and 180° with 2° intervals, produced a set of input vectors with 91 elements. The parameters r and n formed a 2-element output vector. That is, the inverse function was a mapping $f: \mathbf{R}^{91} \rightarrow \mathbf{R}^2$ ($m = 91$, $k = 2$).

For use with data truncated at the forward and backscattering regions network models with correspondingly smaller input vector dimensions m were constructed.

Where indicated, the input vectors were scaled into a 91-dimensional unit hypercube. The linear data input vectors were normalised to give a mean of one over $\theta \in [0°, 180°]$, or the appropriate truncated range, by dividing with a mean taken over all the elements of the input vector. The tests compared several scaling and weighting schemes for the input vectors, including:

1. Logarithmic normalised to 1 at $\theta = 0°$.

2. Linear ($g(\theta) = 1$) scaled into a hypercube.

3. Linear weighted with $g(\theta) = (\sin \theta)^2$.

4. Linear weighted with $g(\theta) = (\sin \theta)^4$.

5. Linear weighted with $g(\theta) = (\sin \theta)^6$.

A two-stage approximation configuration was used to maximise accuracy. The first stage RBF network, trained with data chosen from the following range of parameters: $n \in [1.5, 1.7]$ and $r \in [0.5, 1.5]$ μm, was used for global approximation. Its output was then used to select which local approximation network to use. The second stage networks, performing local approximation, were trained using subsets of the data. Up to 455 pairs (65 radius and 7 refractive index values) of input/output vectors of the data were used to train the first stage, global approximation network. For the second stage the training data was divided into subsets covering 10 overlapping ranges of r, typically with 144 pairs in each subset (there was no subdivision in terms of n).

The number of hidden nodes was chosen to be equal to the number of training samples ($N = S$) in all networks. The i-th input vector was chosen as the centre of the i-th hidden node. The basis function width for the i-th node of RBF network was calculated as the difference between the maximum and minimum distances from the i-th centre vector to all other centres, as defined in Eq. (9). The least squares training method of Eq. (8) was then used. Test data were selected such that all parameter pairs were surrounded by training data pairs in order to eliminate edge effects. All the network algorithms were implemented using the MATLAB™ ver. 4.2 software with the Neural Networks Toolbox ver. 2 (The MathWorks, Inc.).

4. Results and discussion

In the two-stage configuration used in this study the approximation ability of the second-stage, local networks depends on the accuracy with which the first stage, global network classifies the data into correct ranges (in terms of the radius r in for the examples shown in this paper). A failure to classify the data accurately might result in forcing the use of a local network outside the range of output vectors for which it was trained. This would be likely to result in large approximation errors at the second stage. To test the classification ability of the global network, approximation errors were computed for several spatial distributions of training points. It was found that optimum performance was achieved for non-uniform distributions, with decreasing distances between the points for increasing output parameters. In addition, training point density was increased near the boundaries of the local networks. The best results were obtained for input vectors scaled linearly, a $g(\theta) = (\sin \theta)^4$ weighting function and the width constant d of 1.4. Approximation errors for the global network are presented in Fig. 2 as a function of the output parameters. Examination of the results shown in the figure reveals that the lines representing the distance from the test point (true solution) to its approximation generally do not cross network boundaries - in other words, the global network correctly classifies the test data.

The influence of input data scaling and weighting on approximation accuracy was investigated by comparing the five scaling and weighting schemes detailed in the previous section. Local network approximation errors were computed for all the schemes by combining results from 10 sub-networks. Summarised results are shown in Table 1. It can be seen that the best overall result is for linear data weighted with a $g(\theta) = (\sin \theta)^4$ function, closely followed by $(\sin \theta)^6$. In contrast, unweighted input data, both linear and logarithmic, gave poor performance.

Table 1. Relative approximation errors for local networks using five data scaling schemes (results combined from 10 local networks)

Scheme	Scale and weighting	Radius error (%)		Refr. index error (%)	
		Mean	Max.	Mean	Max.
1	log	0.0266	0.8243	0.0253	1.6833
2	linear, hypercube	0.1301	4.0570	0.0395	2.5768
3	linear, $(\sin \theta)^2$	0.0278	1.2762	0.0155	0.8739
4	linear, $(\sin \theta)^4$	0.0016	0.1106	0.0012	0.0765
5	linear, $(\sin \theta)^6$	0.0023	0.2170	0.0016	0.0752

It is interesting to observe that weighting functions of the type $(\sin \theta)^p$ occur in the Gegenbauer transforms which have recently been used to obtain particularly simple representations of angular scattering data and which form the basis of a promising new inversion method.[16,17] The inclusion of the $(\sin \theta)^p$ weighting functions gives emphasis to angular regions where the intensity of light scattered from spherical particles is very sensitive to parameter values, which may account for the results of the present study. Moreover, both the forward- and the back-scattering regions are reduced in magnitude, which is a valuable feature since these regions are absent or distorted in the case of experimental data due to the presence of incident light. Therefore, these results suggest that the common practice of using unweighted linear or logarithmically scaled data in particle light scattering measurements may have to be reassessed.

Relative approximation errors for the combined local networks are shown in Fig. 3 as a function of the output parameters r and n. The existence of high values of approximation error for isolated cases of test data (for example, for $r = 1.3$ and $n = 1.7$ in Fig. 3) stemmed from "anomalous" character of individual test input vectors. In these cases the input vectors were found not to be varying smoothly between the training data points. This finding illustrates a problem common to most neural networks - to minimise errors training examples must adequately represent all cases that the network is to approximate. Therefore, performance can be improved by locally increasing the density of training data. In particular, the large errors along the $n=1.7$ boundary could be reduced quite easily in this way.

Since the forward- and back-scattering regions are absent or distorted in experimental data, the local networks were also trained and tested with incomplete input vectors. Data was removed from 0° up to, but not including, a variable scattering angle, referred to here as a *forward truncation angle*, and/or between a *backscattering truncation angle* and 180°. Approximation errors for such incomplete data are shown in Fig. 4. There was a gradual reduction in accuracy with increasing data loss: up to the forward truncation angle of about 100° or when up to 130° was removed from the backscattering region. After that the reduction was more rapid. While the initial deterioration was slow, the fact that the errors did increase indicated that the truncation of scattering intensity data in angular form resulted in some loss of information. From the practical point of view, however, the accuracy achievable even with a large loss of data was still high. For example, maximum errors were 1% and mean errors were below 0.01% with less than a half of the input data available. An

examination of Figures 4(a) and (b) reveals that, for data weighted with $(\sin \theta)^4$, the forward and backscattering angular regions 0° to about 30° and 120° to 180°, respectively, do not appear to contribute much to the solutions. Moreover, not all of the central 30° to 120° region is required to maintain high accuracy, as shown in Fig. 4(c): 60° to 120° is sufficient. Further tests (not shown) revealed that similar accuracy could be obtained when only the 30° to 60° data region was used.

Typical computation times required for training the networks and particle parameter recovery are shown in Table 2. The evaluations used Matlab 4.2 running on a 166MHz Intel 80586 processor (the training time includes data scaling and normalisation but not the input vector computation from the Lorenz-Mie theory). One global network and 10 local networks, as described in the previous section, were used with complete (91-element) input data vectors scaled accordingly to scheme 4 from Table 1. The parameter area covered by the networks was $0.5\mu m < r < 1.5\mu m$ and $1.5 < n < 1.7$. As can be seen, training the networks took less than a minute. What is more important, size and refractive index recovery took typically only 0.1 second. The computations could be further speeded up by using purpose-written software, truncated data or a faster processor.

Table 2. Computation time for global and local RBF networks.

Phase	Training (s)	Solving (s)
Global [a]	46.8	0.079
Local [b]	10.4	0.023

[a] 455 hidden nodes
[b] 10 subnets, each with 144 nodes

5. Conclusions

Radial basis function neural networks offer a fast and accurate method of solving the inverse scattering problem for small spherical particles. The recovery of both the size and the refractive index of particles from angle-dependent light scattering data is possible. Maximum relative errors of the order of 10^{-3} and mean errors as low as 10^{-5} are easily achievable. To put these results into a more tangible context, the radius errors correspond to absolute errors of less than 0.1nm on average and 2nm maximum for particle diameters. The maximum errors can be further reduced by locally increasing the density of training data. Computations with truncated scattering data show that the absence of scattering data from substantial parts of the 0° to 180° angular range - two thirds or more

- results in only a modest degradation of accuracy. Size and refractive index recovery is fast, taking typically 0.1 second on an average desktop computer. Training the networks is rapid too - for the case presented here it took about a minute. The success with the $(\sin \theta)^p$ weighting function, most notably for $p = 4$, suggests that in experimental work involving angle dependent scattering such weighting may have greater merit than the traditional linear or logarithmic scaling of data.

The parameter space coverage (e.g. the size parameter range) can be extended relatively easily either by simply expanding the range of the training data or, preferably, by adding further local sub-networks (the latter approach would involve a smaller computational penalty). Likewise, extending the present technique to cases involving larger numbers of parameters (for example, complex refractive index, multi-layered spheres, etc.) should not present major difficulties, as long as solutions to the direct problem are available. RBF networks are particularly appropriate in this context, since increasing the number of dimensions should not greatly increase the computational burden. Size distribution recovery may also be possible, although, in common with other methods, accuracy is likely to be dependent on the width and the shape of the distribution.

References

1. P. J. Wyatt, "Some chemical, physical and optical properties of fly ash particles," Appl. Opt. **19**, 975-983 (1980).

2. Z. Ulanowski, "Investigations of microbial physiology and cell structure using laser diffractometry," PhD Thesis, Hatfield Polytechnic, UK (1988).

3. Z. Ulanowski and I. K. Ludlow, "Water distribution, size and wall thickness in *Lycoperdon pyriforme* spores," Mycological Res. **93**, 28-32 (1989).

4. L. A. de Pieri, I. K. Ludlow and W. M.Waites, "The application of laser diffractometry to study the water content of spores of *Bacillus sphaericus* with different heat resistances," J. Appl. Bacteriol. **74**, 578-582 (1993).

5. C. F. Bohren, and D. R. Huffman, *Absorption and scattering of light by small particles* (Wiley, New York, 1983).

6. L. P. Bayvel and A. R. Jones, *Electromagnetic scattering and its applications* (Applied Science Publishers, London, 1981).

7. P. W. Barber and S. C. Hill, *Light scattering by particles: computational methods* (World Scientific, Singapore, 1990).

8. M. R. Jones, B. P. Curry, M. Q. Brewster and K. H. Leong, "Inversion of light scattering measurements for particle size and optical constants: theoretical study," Appl. Opt. **33**, 4025-4034 (1994).

9. G.M. Quist and P. J. Wyatt, "Empirical solution to the inverse scattering problem by the optical strip-map technique," J. Opt. Soc. Am. A **2**, 1979-1985 (1985).

10. V. P. Maltsev and V. N. Lopatin, "Parametric solution of the inverse light-scattering problem for individual spherical particles," Appl. Opt. **36**, 6102-6108 (1997).

11. M. C. Bartholomew-Biggs, Z. Ulanowski and S. Zakovic, "A parameter estimation problem with multiple solutions arising in laser diffractometry," Technical report **281**, Numerical Optimization Centre, University of Hertfordshire, UK (1994).

12. M. C. Bartholomew-Biggs, Z. Ulanowski and S. Zakovic, "Further experience with least squares solutions of an inverse light-scattering problem," Technical report **294**, Numerical Optimization Centre, University of Hertfordshire, UK (1994).

13. S. Zakovic, Z. Ulanowski and M. C. Bartholomew-Biggs, "Application of Global Optimisation to Particle Identification Using Light Scattering," Technical report **325**, Numerical Optimization Centre, University of Hertfordshire, UK (1997).

14. Z. Ulanowski, I. K. Ludlow and W. M. Waites, "Water content and size of bacterial spore components determined from laser diffractometry," FEMS Microbiol. Lett. **40**, 229-232 (1987).

15. J. Everitt and I. K. Ludlow, "Particle sizing using methods of discrete Legendre analysis," Biochem. Soc. Trans. **19**, 504-505 (1991).

16. I. K. Ludlow and J. Everitt, "Application of Gegenbauer analysis to light scattering from spheres: Theory," Phys. Rev. E **51**, 2516-2526 (1995).

17. I. K. Ludlow and J. Everitt, "Systematic behavior of the Mie scattering coefficients of spheres as a function of order," Phys. Rev. E **53**, 2909-2924 (1996).

18. R. Naimimohasses, D. M. Barnett, D. A. Green and P. R. Smith, "Sensor Optimization Using Neural-Network Sensitivity Measures," Measurement Sci. Technol. **6**, 1291-1300 (1995).

19. P. H. Kaye, E. Hirst and Z. Wang, "Neural-network-based spatial light-scattering instrument for hazardous airborne fiber detection," Appl. Opt. **36**, 6149-6156 (1997).

20. A. Ishimaru, R. J. Marks, L. Tsang, C. M. Lam, D. C. Park and S. Kitamura, "Particle-size distribution determination using optical sensing and neural networks," Opt. Lett. **15**, 1221-1223 (1990).
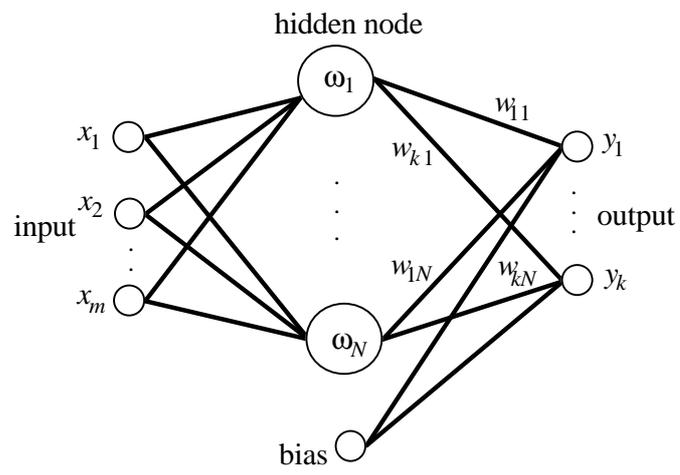
Fig. 1. Architecture of RBF networks.
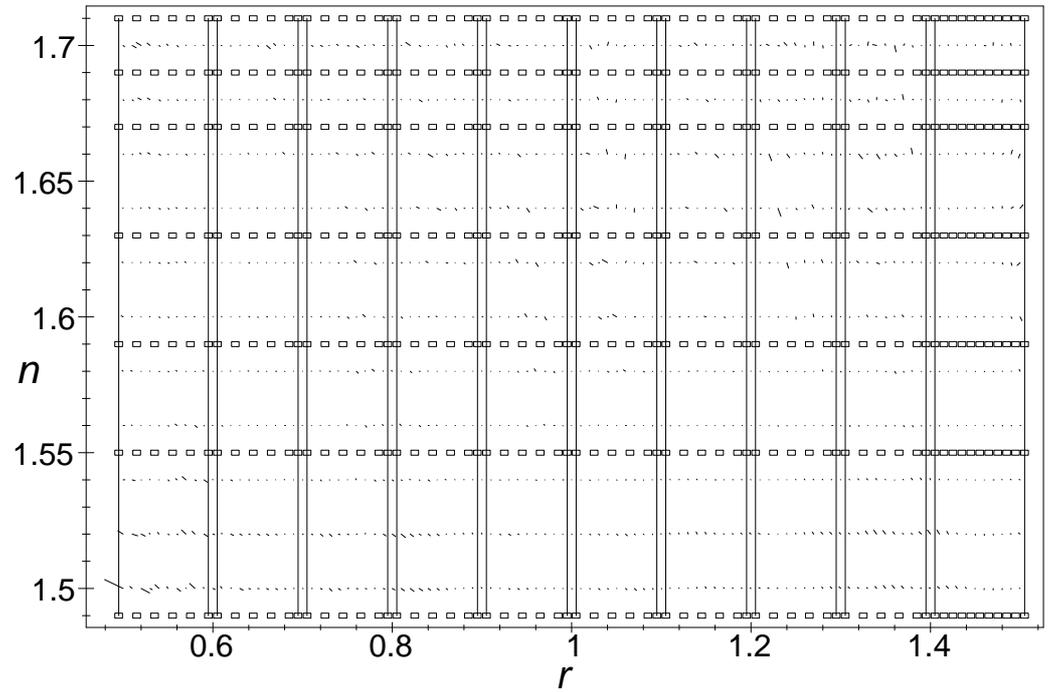
Fig. 2. Approximation errors for the global network shown as a function of the output parameters (radius r and refractive index n). Absolute errors are represented as lines joining the test point and the corresponding approximation (the line length represents the magnitude of the error). The training data points are shown as small rectangles and the boundaries of the 10 overlapping local networks as vertical lines. The input vectors were scaled linearly and the weighting function used was $g(\theta) = (\sin \theta)^4$. The value of the width constant d was 1.4.
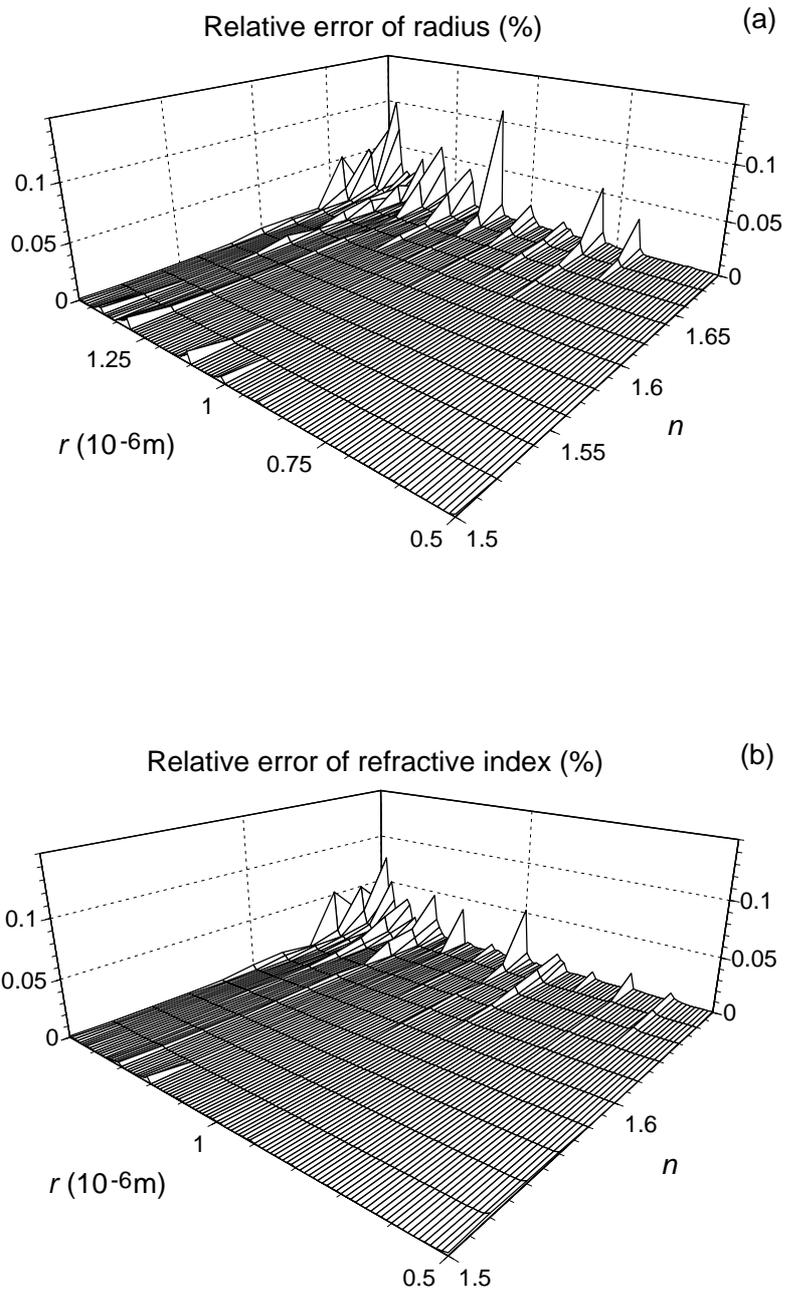
Fig. 3. Approximation errors for the local networks shown as a function of the output parameters r (radius) and n (refractive index). Results from 10 local networks were combined and relative errors calculated separately for r (a) and n (b). The input vectors were scaled linearly, the weighting function used was $g(\theta) = (\sin\theta)^4$ and the width constant d was 1.
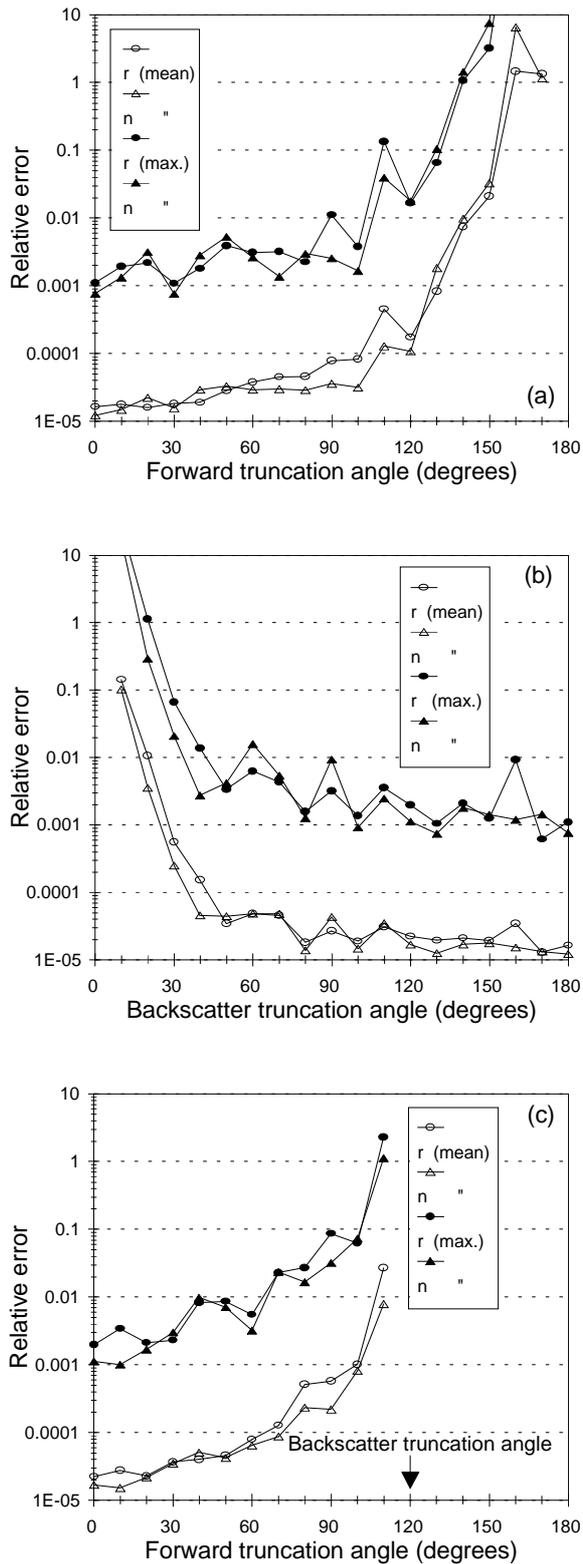
Fig. 4. Approximation errors for the local networks trained and tested with incomplete (truncated) data. The errors are shown as a function of the truncation angle. The data is between the forward scattering truncation angle and 180° in (a), between 0° and the backscattering truncation angle in (b) and between the forward truncation angle and 120° in (c). Mean (open symbols) and maximum relative errors (filled symbols) for the radius (r) and the refractive index (n) are given. The input vectors were scaled linearly, the weighting function used was $g(\theta) = (\sin \theta)^4$ and the width constant d was 1.