

# On Solving the Inverse Scattering Problem with RBF Neural Networks: Noise-Free Case

Z. Wang, Z. Ulanowski and P.H. Kaye

Department of Physical Sciences, University of Hertfordshire, Hatfield, UK

*Neural networks are successfully used to determine small particle properties from knowledge of the scattered light – an inverse light scattering problem. This type of problem is inherently difficult to solve as it is represented by a highly ill-posed function mapping. This paper presents a technique that solves the inverse light scattering problem for spheres using Radial Basis Function (RBF) neural networks. A two-stage network architecture is arranged to enhance network approximation capability. In addition, a new approach to computing basis function parameters with respect to the inverse scattering problem is demonstrated. The technique is evaluated for noise-free data through simulations, in which a minimum 99.06% approximation accuracy is achieved. A comparison is made between the least square and the orthogonal least square training methods.*

**Keywords:** Basis function widths; Function approximation; Inverse light scattering problem; Radial Basis Function neural networks; Small particles

---

## 1. Introduction

Electromagnetic theory can in many cases accurately predict the field produced by scattering from particulate matter, the so-called *direct scattering problem*. Rigorous solutions exist for numerous particle types, such as homogeneous and inhomogeneous spheres, ellipsoids, cylinders, generalised axisymmetric particles, and others. However, far greater practical importance is attached to the determination of par-

ticle properties from knowledge of the scattered fields – the *inverse scattering problem*. Applications that require solutions to the latter problem range from astronomy and remote sensing, through aerosol and emulsion characterisation [1], to non-destructive analysis of single particles and living cells [2–5].

Relatively little progress has been made on the inverse problem, even for the simplest particle shapes. Methods based on approximate models of scattering exist for some particle geometries, for example assuming that the particles are weak (Rayleigh–Debye) scatters or that diffraction alone adequately describes the process. The reader is referred to a large body of existing literature and several monographs for further details [6,7]. However, when such methods are inappropriate, empirical procedures must 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 fitting these solutions to experimental data [8,1–3]. These procedures can be very slow, difficult to implement, and require substantial computing resources. In applying numerical optimisation methods to assist the fitting process, two major difficulties become apparent. First, numerous local solutions necessitate the use of cumbersome and slow global optimisation procedures. Secondly, noise-corrupted or distorted scattering data tends to produce spurious solutions [9].

When developing a method to solve the inverse scattering problem, it is natural to start with the case of a single, homogeneous, non-absorbing spherical particle, and assume a plane incident wave of a known wavelength and state of polarisation in a known medium. The particle can be completely described using two parameters: radius ( $r$ ) and refractive index ( $n$ ). The direct problem can then be solved using the series expansions of Lorenz–Mie

---

*Correspondence and offprint requests to:* Z. Wang, Department of Physical Sciences, University of Hertfordshire, Hatfield, Hertfordshire, UK.

theory [6,7,10]. The present study is confined to such a scattering geometry. Extension to more general cases should be straightforward provided that solutions to the corresponding direct problem are available. In addition, it will be assumed that the light scattered by the particle is measured in only one plane, and presented as a function of the scattering angle (the angle between the direction of the incident light and the line observation). This arrangement leads to a one-dimensional scattering 'pattern' which is dependent on the particle properties, and has previously been used to characterise both single particles and particle distributions [1–3].

Simple least squares fitting of theoretical scattering patterns to measured data often fails to locate correct solutions [2]. Using additional information, such as the positions of the peaks in the light scattering pattern [2,10], or weighting the data in various ways [1], can lead to substantial improvements, although these approaches are still heuristic. Transforming patterns using Legendre or Gegenbauer polynomials allows partial, possibly complete, inversion of scattering patterns for spheres [12–14]. However, for noise-corrupted and incomplete data, the problem remains unsolved. Recent work on determination of particle size distributions from multi-angle light scattering data has shown that neural networks can be used to solve some classes of inverse problems [15].

This paper addresses the issue of solving the inverse scattering problem using neural networks.

One of the most important characteristics of neural networks is the ability to learn input/output relationships from examples. Neural network models, especially Multilayer Feed-Forward (MFF) and Radial Basis Function (RBF) neural networks, have been successfully used for function approximation. Since solving the inverse problem is equivalent to approximating an inverse mapping, it is logical to consider using neural networks. When used for function approximation, the most significant drawback of MFF neural networks is the tendency for the parameter estimation to become trapped at local error minima. In contrast, with a suitable choice of parameters, applying RBF networks to function approximation becomes a linear optimisation problem with no local minima. RBF networks also have the property of best approximation [16,17] and are quickly trained. The theoretical motivation behind using RBF networks includes their link to regularisation theory, and their close relation to Kernel regression estimators [16,18] such that they can also be used as statistical models [19]. From a practical viewpoint, RBF networks are especially suitable for high-dimensional function approximation, since increasing the number of dimensions does not significantly increase the computational burden.

When applying RBF networks to function approximation, the principal difficulties lie in determining the basis function widths. Of existing techniques, *trial and error* is the one most commonly used. The gradient descent method described

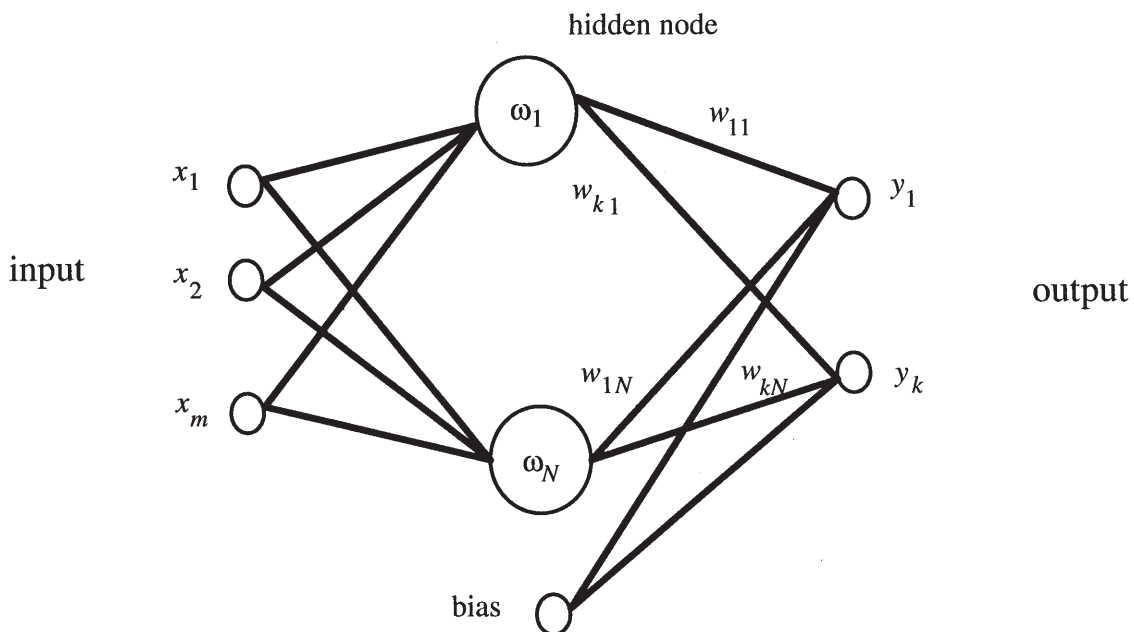


Fig. 1. Architecture of RBF networks.

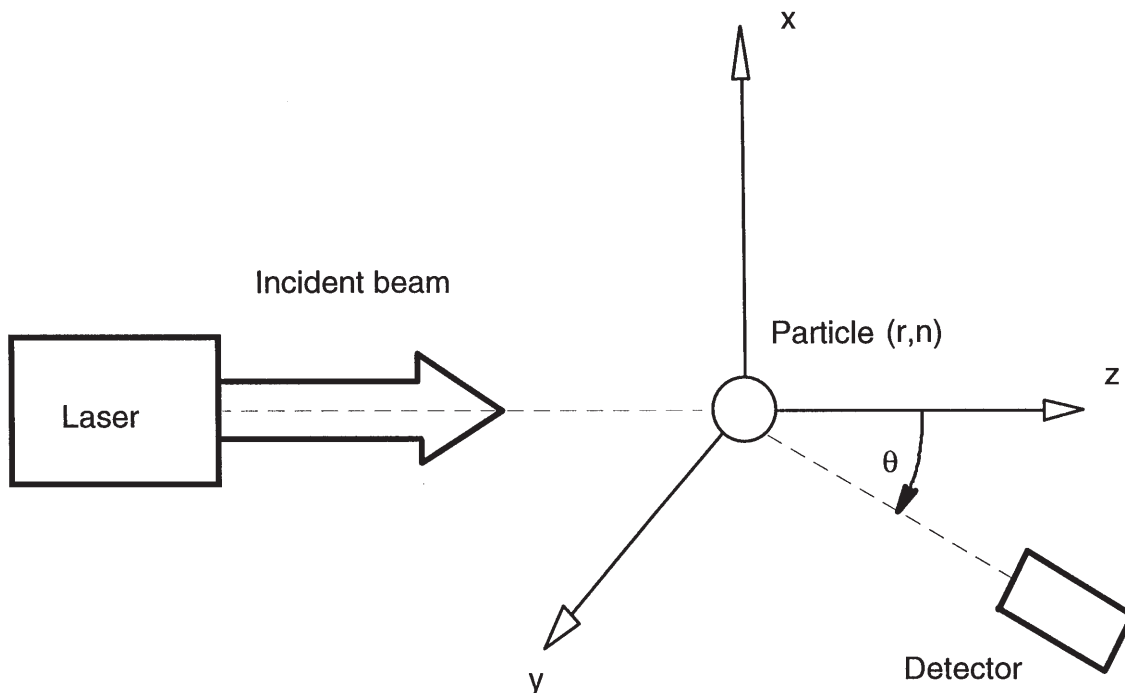


Fig. 2. Scattering geometry.

in Poggio and Girosi [16] risks becoming trapped in local minima. An approach appropriate for classification applications specifies the width by finding the nearest neighbour of the opposite classes [20]. Although Botros and Atkeson have proposed an empirical formula offering reasonably good width approximation using non-linear optimisation [21], calculation for high dimensional problems becomes very costly. A robust width estimation method is presented in Bors and Pitas [22], however, it is only suitable in the application of probability density function estimation. This paper presents a new

approach to computing basis function widths which results in a technique for solving the noise-free inverse scattering problem for spheres with high approximation accuracy and little computation burden. The work has already been successfully extended to noise-corrupted data [19].

## 2. RBF Networks for Function Approximation: Background

A RBF network has a simple architecture consisting of only one hidden layer. The hidden nodes are radial basis functions, and the network output is simply the linear summation of the weighted basis functions. A radial basis function  $\omega(\mathbf{x}, \mathbf{c})$  is a non-linear function solely dependent on the radial distance  $d = \|\mathbf{x} - \mathbf{c}\|$ , where  $\mathbf{c}$  is the function's 'centre'. The implementation of a RBF network includes selecting the basis function widths and centres, and finding the weights. In general, RBF networks have the architecture shown in Fig. 1.

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  $f$ ,  $\bar{f}$ , has the following form corresponding to Fig. 1:

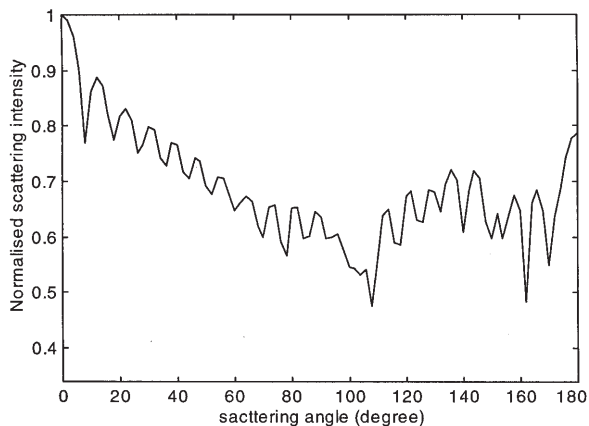


Fig. 3. A scattering pattern which corresponds to  $r = 1.42 \mu\text{m}$  and  $n = 1.7$ . The wavelength of light is  $0.5154 \mu\text{m}$  and the refractive index of the surrounding medium is 1.336.

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

where  $\mathbf{x} \in \mathbf{R}^m$  is the input vector,  $\mathbf{y} \in \mathbf{R}^k$  is the output vector,  $\mathbf{c}_i \in \mathbf{R}^m$  is a centre vector,  $N$  is the number of hidden nodes,  $\mathbf{w}_i \in \mathbf{R}^k$  is a weight vector,  $\mathbf{w}_i = [w_{1i}, w_{2i}, \dots, w_{ki}]^T$ ,  $\omega_i(\mathbf{x}, \mathbf{c}_i)$ s are radial basis functions corresponding to the hidden nodes in Fig. 1, and  $\mathbf{b}$  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. If the basis functions are made Gaussian, the RBF network approximation of  $f$  becomes

$$\mathbf{y} = \sum_{i=1}^N \mathbf{w}_i \exp\{(\mathbf{x} - \mathbf{c}_i)^T \Sigma_i^{-1} (\mathbf{x} - \mathbf{c}_i)\} + \mathbf{b} \quad (2)$$

where  $\Sigma_i$  is a width matrix. The RBF network can be trained by forming the interpolation below with sufficient input and output examples  $(\mathbf{x}_j, \mathbf{y}_j)$ :

$$\mathbf{y}_j = \sum_{i=1}^N \mathbf{w}_i \exp\{(\mathbf{x}_j - \mathbf{c}_i)^T \Sigma_i^{-1} (\mathbf{x}_j - \mathbf{c}_i)\} + \mathbf{b}; \quad j = 1, 2, \dots, S \quad (3)$$

where  $S$  is the number of example input/output pairs. The unknown parameters in this model are the weights  $\mathbf{w}_i$ , the centres  $\mathbf{c}_i$ , and the width matrices  $\Sigma_i$ .

As the interpolations are insensitive to the form of basis function, the widths  $\Sigma_i$  can be chosen randomly. However, for the best generalisation to inputs other than the training examples, RBF networks require optimal  $\Sigma_i$ , which can only be decided through purely supervised training. In spite of that,

it has been stated that purely supervised training offers no advantage over backpropagation, because training becomes a non-linear optimisation problem with local minima [23]. A combined *unsupervised* training method, which only uses input data, and a *supervised* training method, which uses both input and output data, has been suggested by Moody and Darken [23]. First, unsupervised training is used to determine the centres and the simplified width matrices  $\Sigma_i$ . Secondly, supervised training, which becomes a linear optimisation problem, is used to determine the weights. The possibility of choosing suitable centres and widths for the hidden units in order to avoid a non-linear optimisation problem is one of the principal advantages of RBF networks.

Selecting the centres depends upon the required accuracy and how sparse the training data are. Greater accuracy requires more centres, leading to a longer training procedure. When the training data are densely populated, the clustering method is commonly used for selecting the centres [20–22]. However, if the training data are not redundant, then selecting all the training examples is essential.

The width matrices  $\Sigma_i$  define the ellipsoidal receptive fields of the Gaussian functions in multi-dimensional space. In other words, each  $\Sigma_i$  defines a subset of the input which can result in a large output. The width matrices  $\Sigma_i$  control the amount of overlapping and smoothness of the basis functions and, consequently, are associated with the overall neural network generalisation. Determining the width

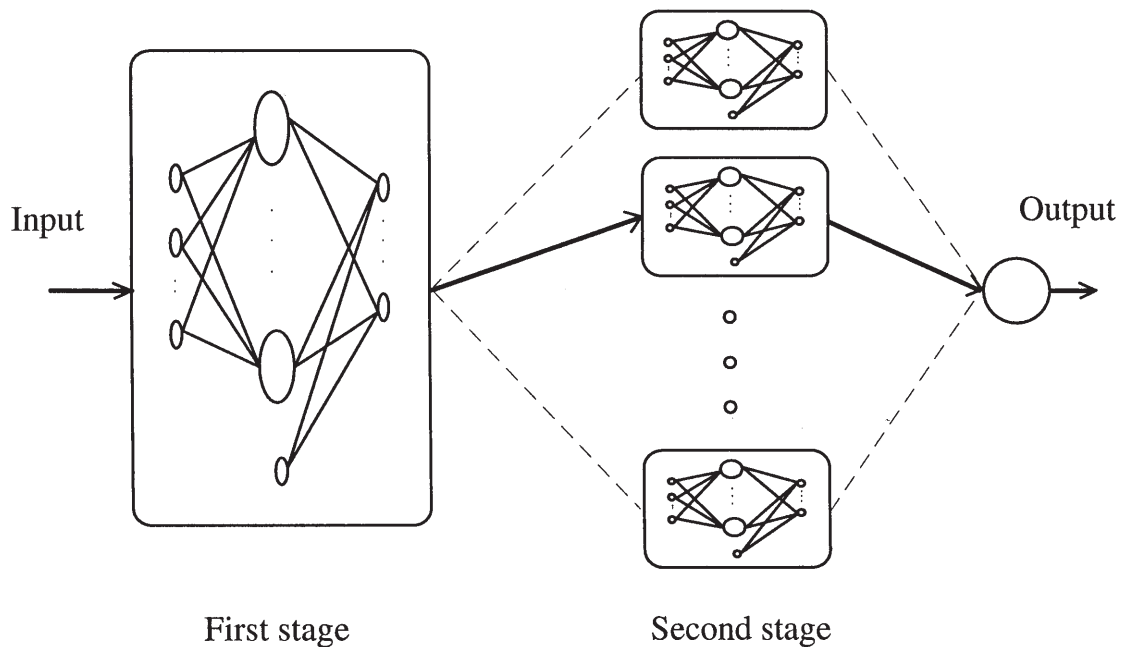
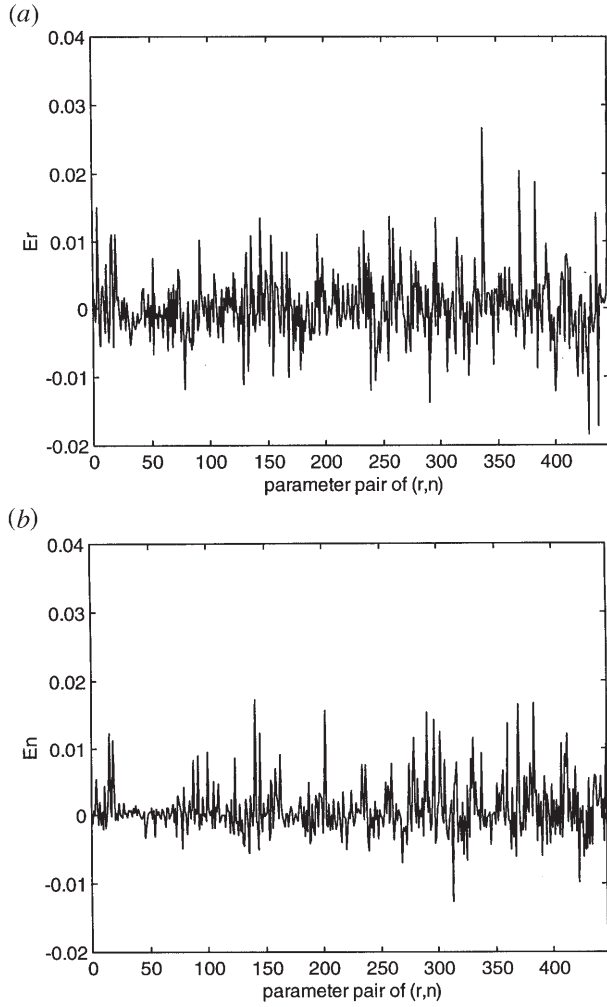


Fig. 4. Architecture of the two stage approximator.



**Fig. 5.** Test errors from the global approximation network. The network was trained with 665 data and tested with 446 data both *type I*. (a) Approximation errors of radius; (b) approximation errors of refractive index.

matrices  $\Sigma_i$  depends upon the application. For classification, the aim is to minimise the overlap between different classes [20]. However, in function approximation, such as this work, sufficient overlapping is required for good network generalisation.

Suppose the centres and the width matrices  $\Sigma_i$  are given. From Eq. (3), training is simply a matter of solving the following linear equation:

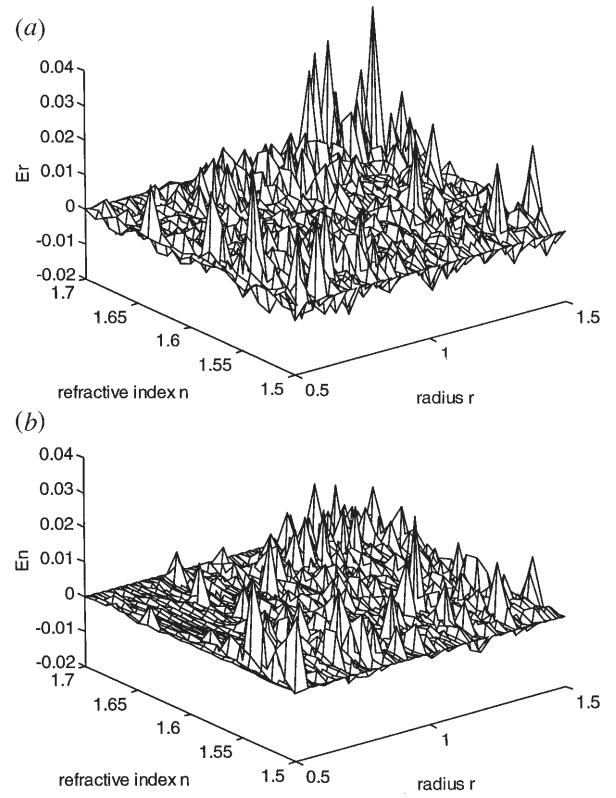
$$[\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_s] = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_N] \cdot \Phi + \mathbf{b} \cdot \mathbf{a} \quad (4)$$

where

$$\Phi = \begin{bmatrix} \phi(1,1) & \phi(1,2) & \dots & \phi(1,S) \\ \phi(2,1) & \phi(2,2) & \dots & \phi(2,S) \\ \dots & \dots & \dots & \dots \\ \phi(N,1) & \phi(N,2) & \dots & \phi(N,S) \end{bmatrix}$$

$$\phi(i,j) = \exp\{(\mathbf{x}_j - \mathbf{c}_i)^T \Sigma_i^{-1} (\mathbf{x}_j - \mathbf{c}_i)\}$$

$$\mathbf{a} = [1 \ 1 \ \dots \ 1];$$



**Fig. 6.** Test errors from the global approximation network. The network was trained with *type I* data and tested with *type II* data. (a) Approximation errors of radius; (b) approximation errors of refractive index.

The bias parameter can be included in the weight matrix to give the form

$$\mathbf{Y} = \mathbf{W}_b \cdot \Phi_a \quad (5)$$

where

$$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_s]$$

$$\mathbf{y}_i = [y_i^1 y_i^2 \dots y_i^k]$$

$$\mathbf{W}_b = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_N, \mathbf{b}]$$

$$\Phi_a = [\Phi^T, \mathbf{a}]^T$$

The network is trained by calculating the least squares error solution of Eq. (5). As a result, the weights are given by

$$\mathbf{W}_b = \mathbf{Y} \Phi_a^T (\Phi_a \Phi_a^T)^{-1} \quad (6)$$

Another supervised training method proposed by Chen [24] is the Orthogonal Least Squares (OLS) approach. However, Sherstinsky and Picard [25] have demonstrated that the OLS training does not produce the smallest possible neural network for a given approximation error. Furthermore, the OLS training is computationally intensive, as shown in Table 1.

### 3. Inverse Scattering Problem: Noise-Free Case

The inverse scattering problem for a homogeneous spherical particle, with respect to a scattering geometry as shown in Fig. 2, can be described as follows: if the 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  $\theta$  on the plane  $z$ - $y$  is a function of the radius and the refractive index  $(r, n)$  of the particle. This function can be given as:

$$I(\theta) = F(\theta, r, n) \quad (7)$$

The function  $F$  is defined by the Lorenz–Mie theory [6,7,10]. Given a vector of discrete measurements  $\mathbf{v}$  of scattered light intensity  $I(\theta)$  at  $m$  different angles

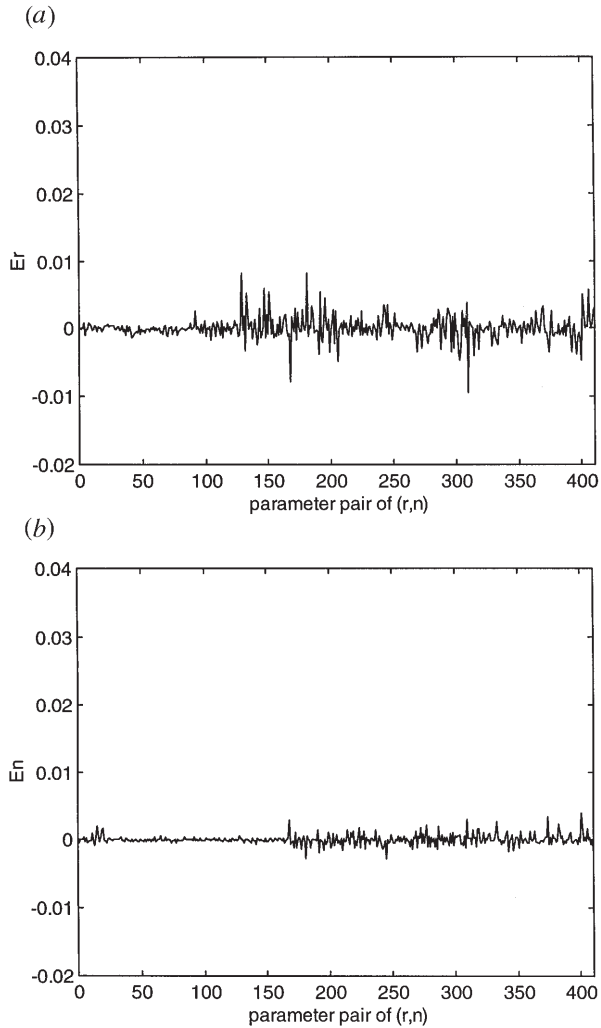
$\theta$ , sufficiently accurate and dense enough to avoid spatial aliasing, the inverse scattering problem is to determine values of  $r$  and  $n$  from the measurements  $\mathbf{v}$ . In other words, it is to find an inverse function  $f = F^{-1}$  such that  $(r, n)$  can be determined by  $f(\mathbf{v})$ . A set of the intensity patterns  $I(\theta)$  and parameters  $r$  and  $n$  generated from Lorenz–Mie theory [6,7,10], are used as the neural network training examples. A typical normalised scattering pattern  $I(\theta)$ , and corresponding parameters are shown in Fig. 3.

A simplified RBF network is used to approximate  $f$  from noise-free data. With the measurement vector  $\mathbf{v}$  as input and  $\mathbf{y}$  representing the parameters  $r$  and  $n$  as output, the simplified RBF network has a form:

$$\mathbf{y} = \sum_{i=1}^N \mathbf{w}_i \exp\left\{-\frac{\|\mathbf{v} - \mathbf{c}_i\|^2}{d_i^2}\right\} + \mathbf{b} \quad (8)$$

where  $d_i$  are the widths. This simplification is obtained by the substitution  $\Sigma_i = d_i^2 \mathbf{I}$  in Eq. (2), such that the ellipsoidal receptive fields of the basis functions become spherical. The motivation behind the use of this model is to reduce the complexity of the width matrices  $\Sigma_i$ . This is particularly relevant to the high-dimensional application considered here. Clearly, the simplification could affect network performance. However, performance can be recovered by scaling the input data to match the changes of the receptive fields. The input training vectors  $\mathbf{v} \in \mathbf{R}^m$  are therefore scaled into a hyper-cube in  $m$ -dimensional space with length 1. With the model of Eq. (8), the inverse scattering problem is solved by first selecting the centres and the widths, followed by supervised training to determine the weights. To maximise accuracy, each training input is chosen to be a centre. As a result, the number of hidden nodes in the RBF network is equal to the number of training input vectors, i.e.  $N = S$  in Eq. (4). This type of network will be called a *complete RBF neural network* within this paper.

Since the widths  $d_i$  control the degree of overlapping of the  $m$ -dimensional Gaussian functions, they are closely associated with the generalisation ability of the RBF network. To obtain good generalisation, the choice of width  $d_i$  should ensure that the  $i$ th basis function responds to the extremes of intensity from its centre  $\mathbf{c}_i$ . This requirement conflicts with the ability of RBF networks to approximate functions locally, which is one reason why they can cope with ill-posed problems. Based on these conflicting principles,  $d_i$  is calculated as the difference between maximum and minimum distances from the centre vector  $\mathbf{c}_i$  to all other training inputs as defined in Eq. (9):



**Fig. 7.** Test errors from the local approximation networks. The 10 networks were tested and trained with *type I* data. Maximum of  $E_r$  was 0.0094 and  $E_n$  was 0.0039. (a) Approximation errors of radius; (b) approximation errors of refractive index.

$$d_i = \max_{j \neq i} \|\mathbf{c}_i - \mathbf{c}_j\| - \min_{j \neq i} \|\mathbf{c}_i - \mathbf{c}_j\|, \quad (9)$$

$$j = 1, 2, \dots, S$$

Consequently, the wider the data spread, the larger the width is and the network is automatically adjusted to different input data.

#### 4. Simulation

Simulations were conducted using theoretical data generated from Lorenz–Mie theory. Discrete ranges of parameter values,  $n \in [1.5, 1.7]$  (dimensionless), and  $r \in [0.5, 1.5]$  (in micrometers), were used to construct a matrix of parameter pairs from all combinations. In addition, a wavelength of  $0.5145 \mu\text{m}$  and a refractive index of the surrounding medium (water) of 1.336 were assumed. Each parameter pair had a corresponding normalised log intensity pattern  $I(\theta)$ ,  $\theta \in [0, 180]$  which was calculated by

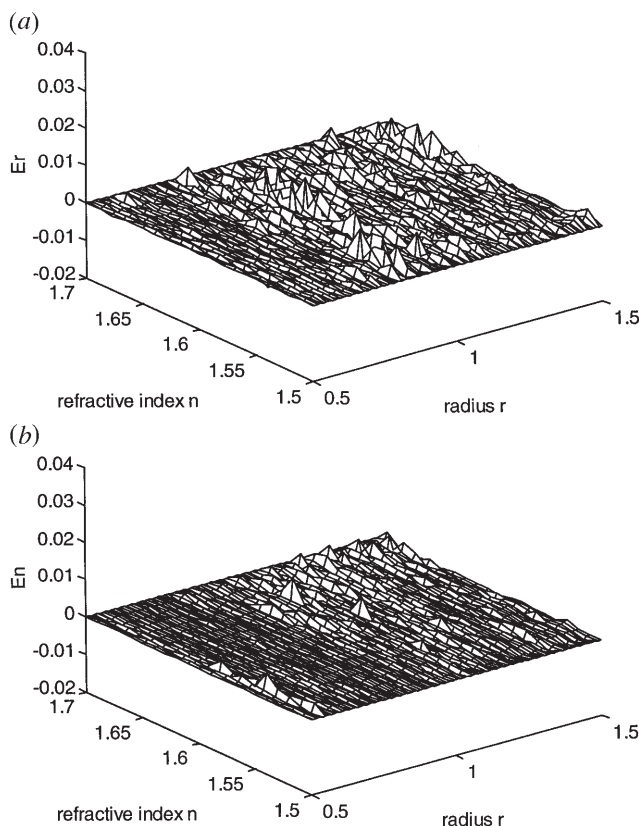
$$I(\theta) = 1 - \frac{\log|S_1(0)|^2 - \log|S_1(\theta)|^2}{10} \quad (10)$$

where  $S_1(\theta)$  is an element of the scattering matrix

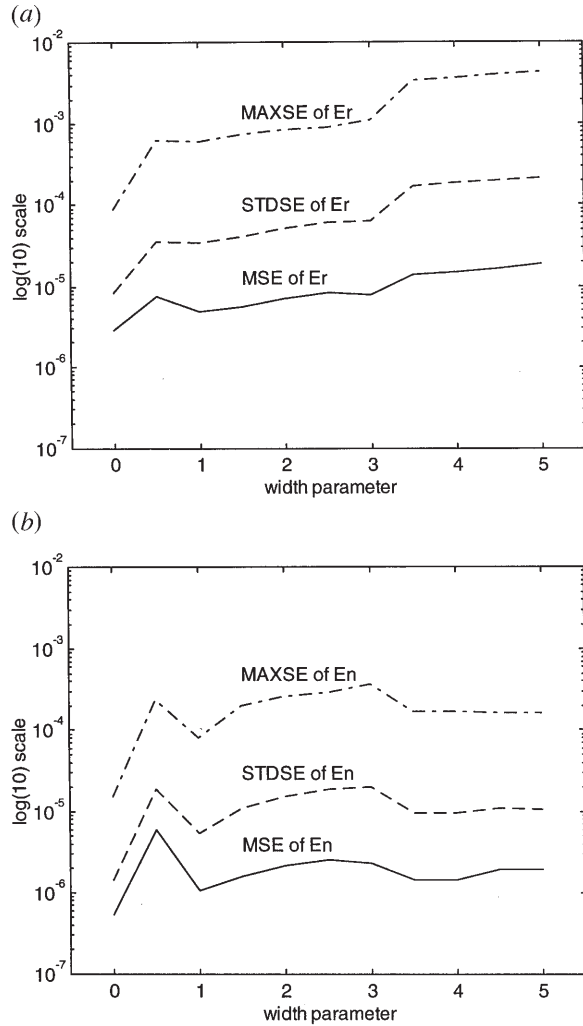
[6,8]. Note that, since  $|S_1(\theta)|^2 \leq |S_1(0)|^2$ ,  $I(\theta) \leq 1$  and  $I(0) = 1$ . The intensity patterns, sampled between  $0^\circ$  and  $180^\circ$  with  $2^\circ$  intervals, produced input training vectors with 91 elements, which were then scaled into a hyper-cube in 91-dimensional space with length 1. Parameters  $r$  and  $n$  formed a 2-element output vector. That is, the inverse function is a mapping  $f: \mathbf{R}^{91} \rightarrow \mathbf{R}^2$ . Two types of data were generated with different increments of refractive index: *type I* were generated with increments of 0.02, and *type II* with increments of 0.01. The radius was incremented by  $0.01 \mu\text{m}$ . Since the parameter ranges are the same for both types of data, *type I* data files contain 1111 pairs and *type II* contain 2121 pairs of inputs and outputs.

Preliminary experiments showed that if the increment between parameters was kept constant, the approximation errors were found to be roughly proportional to the parameter range. This can be seen in the substantial reduction of errors from Figs 5–8. A practical two stage approximation configuration was therefore designed with an architecture as shown in Fig. 4.

The first stage RBF network, trained with data chosen from the full range of parameters:



**Fig. 8.** Test errors from the local approximation networks. The networks were trained with *type I* data and tested with *type II* data. Maximum of  $E_r$  was 0.0094 and  $E_n$  was 0.0060. (a) Approximation errors of radius; (b) approximation errors of refractive index.



**Fig. 9.** The Mean of Square Errors (MSE), Standard Deviation of Square Errors (STDSE) and MAXimum of Square Errors (MAXSE) are compared for the calculated widths and a range of constant widths. The  $x$  axis shows the width parameters from 0.5 to 5 with 0.5 interval. Results plotted against '0' width on  $x$  axis correspond to the widths calculated by Eq. (9). The least squares method was used to train networks. The *type I* data were used for both training and testing. (a) Approximation errors of radius; (b) approximation errors of refractive index.

$n \in [1.5, 1.7]$  (dimensionless) and  $r \in [0.5, 1.5]$  ( $\mu\text{m}$ ), was used for global approximation. Its approximated output is then used to select which local approximation network to use. The second stage networks perform local approximation and are trained with data chosen from small subsets of the *type I* data.

#### 4.1. Training

- The intensity patterns  $I(\theta)$  were used as input and the parameters  $(r, n)$  used as output to train the network.

- 665 pairs of input/output vectors of *type I* data were used to train the first stage, global approximation network. The training data were selected such that all the test data were surrounded by training data. All training inputs were initially scaled.
- *Type I* data was divided into 10 subsets for the second stage training and testing. Within each subset, training data were selected in the same way as for the global approximation. All training inputs were also initially scaled.
- All the RBF networks were complete, i.e. the number of hidden nodes was chosen to be equal to the number of training samples. 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. (6) was used in this application because it proved faster and more accurate than the OLS training method, as shown in Table 1.

#### 4.2. Testing

- For each test pattern, the approximated parameters: radius  $r'$  and refractive index  $n'$  were calculated using Eq. (8). The approximated parameters  $(r', n')$  were then compared with the actual parameters  $(r, n)$  by defining the relative test error Er and En:
 
$$\text{Er} = \frac{r - r'}{r}; \quad \text{En} = \frac{n - n'}{n} \quad (10)$$
- To test the first stage approximation, *type I* data, excluding the training data, was used. The test results, Er and En, are shown in Fig. 5. All *type II* data were also used to test the first stage approximation, and the results are shown in Fig. 6.
- To test each local approximation network, the corresponding subset of *type I* data, exclude training data was used. The test results are shown in Fig. 7. All *type II* data were also used to test local approximation, and the results are shown in Fig. 8.
- The maximums of Er and En for global approximation with *type I* data, as shown in Fig. 5, were as high as 4%. In contrast, the maximums of Er and En for local approximation with *type I* data, as shown in Fig. 7, were about four times smaller

**Table 1.** Comparison between different training methods.

Square errors ( $\times 10^5$ )		Least Squares training method		OLS training method	
		<i>type I</i>	<i>type II</i>	<i>type I</i>	<i>type II</i>
MSE <sup>a</sup>	Er	0.2861	0.1265	7.552	6.218
	En	0.0538	0.0315	0.771	0.560
STDSE <sup>b</sup>	Er	0.8382	0.4698	40.72	96.10
	En	0.1455	0.1402	3.05	6.52
MAXSE <sup>c</sup>	Er	8.79	8.79	500	4080
	En	1.522	3.614	40	260
Training CPU time		3'28"		46'29"	

<sup>a</sup>Mean Square Error<sup>b</sup>Standard Deviation of Square Error<sup>c</sup>Maximum of Square Error

for the radius and six times smaller for the refractive index. Clearly, the global approximation alone is much less accurate. However, when using only local approximation, prior knowledge about the particle parameters is required to choose a suitable network. Therefore, the design with combined two stage approximation architecture, as shown in Fig. 4, has been justified.

- The mean of square test errors, standard deviation of square test errors, and maximum of square test errors for *type I* and *type II* data from final local approximation networks are listed in Table 1.
- All the test results displayed in Figs 5–8 were produced from networks trained by the least squares method of Eq. (6).

A comparison was made between the results obtained using RBF network with widths calculated by Eq. (9) and constant widths. The networks were trained and tested using *type I* data for the local approximation networks. The test results are shown in Fig. 9. The comparison in Fig. 9 indicates that the networks using widths calculated by Eq. (9) produced the best results. Furthermore, the network does not rely on trial and error.

Table 1 compares OLS and least squares training by Eq. (6) with results from local approximation networks. Both methods were used to train with the *type I* data. The *type I* test data and all of the *type II* data were used to test both methods. The comparison values are based on the square errors of each test data. All errors shown in Table 1 are scaled by  $10^5$ . The comparison in Table 1 demonstrates that the OLS training method is not only computationally intensive, but also gives poor generalisation.

In summary, simulations have shown that:

1. RBF networks can be used as an approximation

model for solving the inverse light scattering problem. The proposed two stage approximation constructed from *complete RBF neural networks* offers a practical approach to solving the inverse scattering problem with particle sizes and refractive indices covering a wide range of interest.

2. The *complete RBF network* model, with all input training vectors as the centres, using the width parameters calculated by Eq. (9), can consistently generalise well.
3. The advantages of using the widths calculated by Eq. (9) are: accuracy, ease of implementation and reduced computation. The disadvantage is that the input data need appropriate scaling.
4. Constant widths can be used when accuracy is less critical, and a trial and error approach is acceptable.
5. Networks trained with carefully chosen training data approximated better.

In demonstrating how RBF neural networks can be used to solve the inverse scattering problem, noise-free data has been assumed in this paper. Extension to noise-corrupted data is presented in Wang and Ulanowski [19].

## 5. Conclusion

An approximation technique using RBF networks has been presented for solving an inverse scattering problem. It has been shown that the two stage approximation architecture constructed from *complete RBF neural networks* offers better approximation ability. Along with a new unsupervised approach, developed to calculate the basis function widths, the two stage RBF neural networks configuration achieved good approximations (minimum

99.06% accuracy) from noise-free data. This work has demonstrated the ability of RBF networks to approximate high dimensional non-linear functions, commonly an ill-posed problem, and more importantly, it has opened up an area associated with solving the inverse light scattering problem.

**Acknowledgements.** The authors would like to thank Dr Marc Thomas, Dr Alistair Ferguson and Professor Lawrence Dixon for their invaluable suggestions and time spent on presenting this work accurately.

## References

- Wyatt PJ. Some chemical, physical and optical properties of fly ash particles. *Applied Optics* 1980; 19(6): 975–983
- Ulanowski Z. Investigations of microbial physiology and cell structure using laser diffractometry. PhD Thesis Hatfield Polytechnic, 1988
- Ulanowski Z, Ludlow IK. Water distribution, size and wall thickness in *Lycoperdon pyriforme* spores. *Mycological Res* 1989; 93: 28–32
- de Pieri LA, Ludlow IK, Waites WM. The application of laser diffractometry to study the water content of spores of *Bacillus sphaericus* with different heat resistances. *J Applied Bacteriology* 1993; 74: 578–582
- Ludlow IK, Kaye PH. Scanning diffractometer for the rapid analysis of micro particles and biological cells. *J Colloid and Interface Sci* 1978; 69: 571–589
- Bohren CF, Huffman DR. Absorption and scattering of light by small particles. Wiley-Interscience, New York, 1983
- Bayvel LP, Jones AR. Electromagnetic scattering and its applications. Applied Science Publishers, London, 1981
- Quist GM, Wyatt PJ. Empirical solution to the inverse-scattering problem by the optical strip-map technique. *J Optical Soc Am A* 1985; 2 (11): 1979–1985
- Bartholomew-Biggs MC, Ulanowski Z, Zakovic S. A parameter estimation problem with multiply solutions arising in laser diffractometry. Technical Report 281, Numerical Optimisation Centre, University of Hertfordshire, 1994
- Barber PW, Hill SC. Light scattering by particles: computational methods. World Scientific, Singapore, 1990
- Ulanowski Z, Ludlow IK, Waites WM. Water content and size of bacterial spore components determined from laser diffractometry. *FEMS Microbiology Lett* 1987; 40: 229–232
- Everitt J, Ludlow IK. Particle sizing using methods of discrete Legendre analysis. *Biochemical Soc Trans* 1991; 504–505
- Ludlow IK, Everitt J. Application of Gegenbauer analysis to light scattering from spheres: *Theory Phys Rev* 1991; E 51 (3B): 2516–2526
- Ludlow IK, Everitt J. Systematic behaviour of the Mie scattering coefficients of spheres as a function of order, *Phys Rev* 1996; E 52 (3): 2909–2924
- Naimimohasses R, Barnett DM, Green DA, Smith PR. Sensor optimisation using neural-network sensitivity measures. *Meas Sci & Tech* 1995; 6 (9): 1291–1300
- Poggio T, Girosi F. Networks for approximation and learning, *Proc IEEE* 1990; 78(9): 1481–1497
- Girosi F, Poggio T. Networks and the best approximation property, *Biol Cybern* 1990; 63: 169–176
- Xu L, Krzyzak A, Yuille A. On radial basis function nets and kernel regression: statistical consistency, convergence rates, and receptive field size, *Neural Networks* 1994; 7 (4): 609–628
- Wang Z, Ulanowski Z. On solving the inverse scattering problem with RBF neural networks: noise-corrupted case, Technical Report, ERDC University of Hertfordshire 1996
- Musavi MT, Chan KH, Hummels DM. On the training of radial basis function classifiers, *Neural Networks* 1992; 5: 595–603
- Botros S, Atkeson CG. Generalisation properties of radial basis functions. In: Lippmann RP, Moody JE, Touretzky DS (eds), *Advances in Neural Information Processing System 3*, Morgan Kaufmann, San Mateo, 1991, pp 707–713
- Bors AG, Pitas I. Robust estimation for radial basis functions. In: *Neural Networks for Signal Processing IV*, Proc 1994 IEEE Workshop, pp 105–114
- Moody J, Darken CJ. Fast learning in networks of locally-tuned processing units. *Neural Computing* 1989; 1: 281–294
- Chen S, Cowan FN, Grant PM. Orthogonal least squares learning algorithm for radial basis function networks, *IEEE Trans Neural Networks* 1991; 2(2): 302–309
- Sherstinski A, Picard RW. On the efficiency of the orthogonal least squares training method for radial basis function networks. *IEEE Trans Neural Networks* 1996; 7 (1): 195–200

## Nomenclature

$\omega(\ )$	basis function
$\mathbf{c}$	centre vector
$\mathbf{x}$	input vector
$\mathbf{y}$	output vector
$\mathbf{R}^m$	$m$ -dimensional space
$f$	function
$\mathbf{w}_i$	$i$ th weight vector
$w_{ij}$	element of weight matrix
$\mathbf{b}$	bias vector
$\Sigma_i$	width matrices
$\Sigma$	symbol of summation
$\mathbf{a}$	identity vector
$\mathbf{W}$	weight matrix
$\Phi$	matrix
$\mathbf{y}_j$	input training vector
$\mathbf{x}_j$	output training vector
$I(\theta)$	intensity function
$\mathbf{v}$	measurement of intensity function
$\theta$	scattering angle
$d_i$	$i$ th width parameter
$\log$	logarithm
$S_1(\ )$	element of the scattering matrix
$E$	relative test error
$r$	radius parameter
$n$	refractive index parameter