Artificial Intelligence in Medicine (2004) 31, 211–220

ELSEVIER

ARTIFICIAL INTELLIGENCE IN MEDICINE

http://www.intl.elsevierhealth.com/journals/aiim

Comparison of neural network predictors in the classification of tracheal–bronchial breath sounds by respiratory auscultation

Ross Folland[a,*], Evor Hines[a], Ritaban Dutta[a], Pascal Boilot[a], David Morgan[b]

[a]*Intelligent Systems Engineering Laboratory, Electrical and Electronics Division, School of Engineering, University of Warwick, Coventry CV4 7AL, UK*
[b]*Birmingham Heartlands Hospital, Bordesley Green East, Birmingham B9 5SS, UK*

Received 23 April 2003; received in revised form 6 November 2003; accepted 17 January 2004

KEYWORDS
Breath sounds;
Neural network;
Auscultation;
CPNN; MLP;
RBFN

Summary Despite extensive research in the area of identification and discrimination of tracheal–bronchial breath sounds by computer analysis, the process of identifying auscultated sounds is still subject to high estimation uncertainties. Here we assess the performance of the relatively new constructive probabilistic neural network (CPNN) against the more common classifiers, namely the multilayer perceptron (MLP) and radial basis function network (RBFN), in classifying a broad range of tracheal–bronchial breath sounds. We present our data as signal estimation models of the tracheal–bronchial frequency spectra. We have examined the trained structure of the CPNN with respect to the other architectures and conclude that this architecture offers an attractive means with which to analyse this type of data. This is based partly on the classification accuracies attained by the CPNN, MLP and RBFN which were 97.8, 77.8 and 96.2%, respectively.

We concluded that CPNN and RBFN networks are capable of working successfully with this data, with these architectures being acceptable in terms of topological size and computational overhead requirements. We further believe that the CPNN is an attractive classification mechanism for auscultated data analysis due to its optimal data model generation properties and computationally lightweight architecture.
© 2004 Elsevier B.V. All rights reserved.

1. Introduction

When exposed to a disease process, be it a foreign organism or a more benign source, the respiratory system's natural defences are effected. Primarily, these defences consist of mucous membranes that secrete in order to protect the lower respiratory system from foreign bodies. Auscultation forms a major, but not exclusive, part of the examination of the respiratory system [1]. Dependent upon the physical location of the disease process and the consequent changes in the lung tissue, audible changes can be noted during respiration. By auscultating the chest, the medical examiner can determine the location of the disease process by the respiratory sounds produced by the patient. Combined with a knowledge of respiratory pathology, the practitioner can diagnose the probable disease process and appropriately medicate the patient. Not only is auscultation of the chest a fundamental

* Corresponding author. Tel.: +44-2476-528-146; fax: +44-2476-418-922.
E-mail address: r.s.folland@warwick.ac.uk (R. Folland).

0933-3657/$ – see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.artmed.2004.01.004

procedure for assessing respiratory competence, but it is also used for assessing the patient's general physiological state. However even in skilled hands auscultation can be an unreliable diagnostic manoeuvre. A study in Seattle showed that clinical examination has a sensitivity of about 50% and specificity between 60 and 70% in diagnosing pneumonia [2].

Studies in the field of artificial neural networks (ANNs) and their application to the detection of specific types of respiratory sounds have previously shown promising results [3–5]. However, in these papers particular attention has been given to analysing a certain subset of tracheal–bronchial sounds, such as asthma or pneumonia. Furthermore a common ANN employed throughout those studies was the multi layer perceptron (MLP). In this paper we evaluate the application of constructive probabilistic neural networks (CPNNs) to the analysis and detection of the spectrum of tracheal–bronchial breath sounds encompassing wheezes, crackles and stridor. Performance of the CPNN is assessed in comparison with the MLP and radial basis function network (RBFN). In addition, a comparison is also made with the results obtained from expert medical examiners. In the following section, neural network architectures under consideration are introduced; Section 3 describes the data processing stages prior to neural network analysis; Section 4 details the results obtained from the neural network implementations; and Sections 5 and 6 discuss and concludes on our results.

2. Neural network classifiers

ANNs are biologically inspired statistical mechanisms frequently employed in tasks requiring pattern recognition and classification. Due to the inherent complexity of pattern classification, various ANN architectures exist that attempt to reconcile the problems found in pattern recognition. Here we have focused on MLP, RBFN and CPNN ANN architectures, and these are described in more detail below.

2.1. Introduction to selected neural network paradigms

MLPs, one of the most popular type of ANNs, are feed-forward networks of simple processing elements or neurons whose connectivity resembles that of the brain [6,7]. Each neuron performs a weighted sum of its inputs and transforms the result through a typically non-linear activation function. An MLP is able to learn arbitrarily complex non-linear regressions by adjusting its connective weights using a training algorithm such as back propagation gradient descent with momentum [7,8]. At each stage in the training, the MLP processes all of its inputs in a feed-forward manner. It compares the resulting outputs with the desired target outputs and back-propagates any errors, adjusting each weight in the network according to its contribution to the overall error. The complexity of the network can be controlled by a priori determination of the optimal number of neurons forming the hidden layer considering various factors (e.g. data complexity, level of noise, training algorithm, etc.). It can be shown that the MLP attempts to estimate the Bayesian class-conditional probability densities $P(\mathbf{x}|C_k)$ and hence tries to determine optimal hyperplane discriminant functions for partitioning feature space [8].

RBFNs are feed-forward connectionist architectures consisting of a hidden layer of radial kernels and an output layer of linear neurons [6,7]. Although the architecture of RBFNs resembles that of MLPs, their input-output mappings and training algorithms are fundamentally different. Each hidden neuron in an RBFN is tuned to respond to a local region of feature space by means of a radially symmetric Gaussian function. It can be seen that the radial basis functions form Gaussian component densities that are used in conjunction with a Gaussian mixture model (GMM). The result of this GMM is a system that similarly attempts to estimate the $P(\mathbf{x}|C_k)$ functions. RBFNs are typically trained using a hybrid algorithm that employs unsupervised learning for the hidden layer followed by supervised learning of the output layer. The first step consists of selecting the radial basis centres and spreads following the forward stepwise selection orthogonal least squares algorithm, which generates an appropriate number of hidden neurons. In the second step, training the output is a straightforward supervised problem, in which the radial basis activations are used as regressors to predict the class target outputs. With this training algorithm, a value of the spread constant which induces the creation of a number of radially symmetric homoscedastic Gaussian functions has to be defined.

In summary, MLPs partition the feature space with hyper-planes whereas the RBFN's decision boundaries are typically hyper-spheres derived from Gaussian functions with circular probability contours partitioning feature space with non-linear discriminants. CPNNs, however, partition the feature space with hyper-ellipsoids of *varying* sizes (i.e. hyper-ellipsoidal probability contours) which is discussed in the next section.

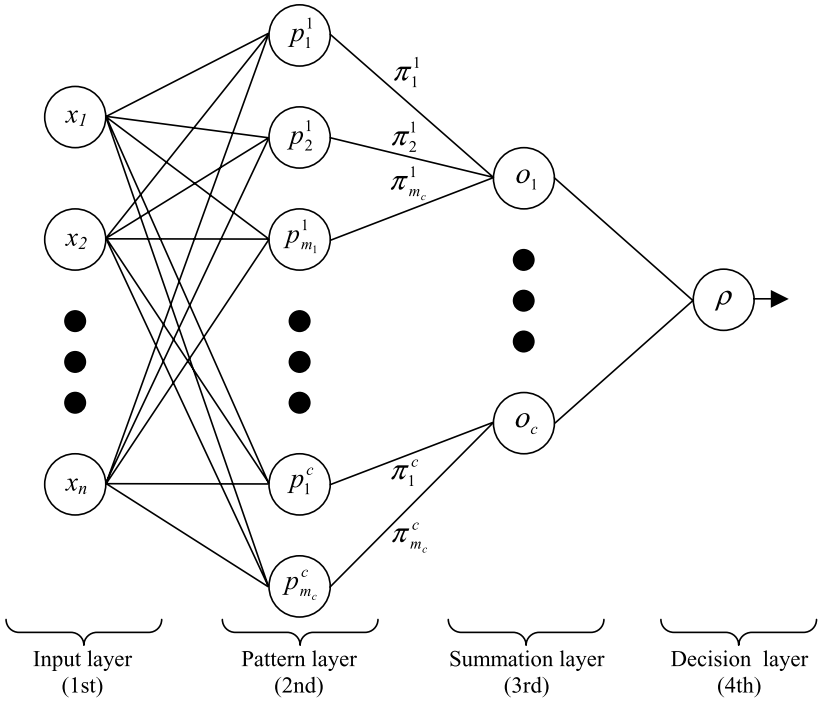

Figure 1 Basic design of a probabilistic neural network.

2.2. The constructive probabilistic neural network

A further neural network architecture is the probabilistic neural network (PNN) [9]. The PNN works by creating a set of multivariate probability densities that are derived from the training vectors presented to the network. The first layer of the PNN (Fig. 1) simply presents the input patterns to the network. These input vectors are propagated to the pattern layer where a new neuron (i.e. pattern unit p_j^k) is created, having a mean vector equal to the input pattern and is hence receptive to future occurrences of this pattern. The densities for each pattern unit j of class k are calculated in the second layer and is then used in the summation (third) layer to estimate the class-conditional probability densities $P(\mathbf{x}|C_k)$. The fourth layer applies Bayes decision rules [10,11] to the $P(\mathbf{x}|C_k)$ of class C_k in computing the output classification. The PNN similarly creates Gaussian functions (of circular probability contours) in feature space that represent the available training data. Again, the PNN employs GMM to estimate the posterior probabilities.

Due to each pattern layer Gaussian component density $p(\mathbf{x})$ being derived from one training vector, the PNN is limited to applications involving relatively small datasets [12]. Large datasets would lead to large network architectures, which would have an adverse impact on computational complexity. In addition, this could saturate the feature space with overlapping Gaussian functions that would increase the rate of misclassification. Furthermore, the training algorithm employs a homoscedastic smoothing function σ across all of the pattern layer's component densities. This prevents the PNN from individually tuning the pattern units to each new training vector, consequently limiting the ANN's generalisation and specificity capabilities.

The Constructive Probabilistic Neural Network (notionally introduced by Streit and Luginbuhl [13]) using the *dynamic decay adjustment* algorithm is architecturally similar to the PNN with the advantage that it is grown as a result of a particular training algorithm. Again the CPNN uses GMM in its estimation model but the algorithm used to introduce and tune the component densities differs insofar that it evaluates existing Gaussian mixtures in determining whether additional neurons are required. The CPNN is presented with an n-dimensional vector $\mathbf{x}$ on the first layer. In the second layer, each pattern unit estimator j of class k computes the multivariate Gaussian probability density using

$$p_j^k(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}|\Sigma_j^k|^{-1/2}} \times \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_j^k)^{\mathsf{T}}(\Sigma_j^k)^{-1}(\mathbf{x}-\mu_j^k)\right\} \quad (1)$$

where μ_j^k is the mean vector and Σ_j^k is the covariance matrix. As can be seen in (1) the scaling term to the left of the exponential is such that $\int_{-\infty}^{\infty} p(\mathbf{x})\,d\mathbf{x} = 1$.

Furthermore, $p(\mathbf{x})$ is governed by the Mahalanobis distance measure $-(1/2)(\mathbf{x}-\mu_j^k)^{\mathrm{T}}(\Sigma_j^k)^{-1}(\mathbf{x}-\mu_j^k)$ given in the exponent. The class estimation density $o_k(\mathbf{x})$ is then computed in the third layer by aggregating the product of the pattern component densities with a Gaussian mixing proportion π_j^k in a GMM framework, simply expressed as

$$o_k(\mathbf{x}) = \sum_{j=1}^{m_k} \pi_j^k p_j^k(\mathbf{x}), \quad \text{where } \sum_{j=1}^{m_k} \pi_j^k = 1, k = 1, \ldots, K \tag{2}$$

where m_k is the number of pattern units in class k. The output layer applies the Bayes decision rule to the computed mixed estimation densities. The decision rule is applied such that

$$D(\mathbf{x}) = n, \quad \text{when } \forall m \neq n : h_n c_n f_n(\mathbf{x}) > h_m c_m f_m(\mathbf{x}) \tag{3}$$

where $D(\mathbf{x})$ is the classification, h_n is the prior probability of the occurrence of pattern $\mathbf{x}$ in class n, c_n is the cost function of erroneously classifying pattern $\mathbf{x}$ in a different class and $f_n(\mathbf{x})$ is the class posterior probability density [10,11]. By setting the cost functions to 1 and by ensuring equal representation of all class patterns in the dataset such that for all n $h_n = 1$, the Bayes classification reduces to the criterion of largest class posterior probability density, i.e. $D(\mathbf{x}) = n$ when $f_n(\mathbf{x}) > f_m(\mathbf{x})$ $(\forall m \neq n)$.

A salient advantage of the CPNN over the PNN is that it readily implements heteroscedastic smoothing functions σ_j^k over the pattern units' estimators instead of employing a global function. Comparable architectures such as the RBFN would require modifications to the training algorithm to achieve this. This allows the CPNN to adapt each pattern unit's kernel shape-dependent upon the data presented to the network thereby permitting the use of ellipsoidal representations, i.e. kernels of ellipsoidal probability contours.

The CPNN architecture is grown according to the features present in the training dataset. The dynamic decay adjustment (DDA) algorithm [12] permits the discrimination between supporting and conflicting pattern units in the architecture through an assessment of each pattern unit's component density function. This is achieved by introducing two threshold functions θ^+ and θ^- that specify the minimum probability required for correct classification and the maximum probability for misclassification, respectively. The disparate region between the two thresholds defines the region of conflict, in which an ambiguous classification is made which can usually be rejected on the grounds of low confidence. Consequently the function of the DDA algorithm is two-fold; firstly to ensure that the rate of misclassification is reduced by creating new locally receptive pattern units when necessary, i.e. by introducing new neurons; and secondly, reducing the number of training patterns falling within the region of conflict by adjusting the pattern units' component density function. After presenting all input–output pairs ($\mathbf{x}$, k) to the CPNN, Berthold and Diamond [12] state that two conditions must hold: (1) At least one pattern unit in the correct class k has a function response greater than or equal to θ^+,

$$\exists i, \quad 1 \leq i \leq m_k : p_i^k(\mathbf{x}) \geq \theta^+ \tag{4}$$

and (2) that all conflicting pattern units (i.e. all other-class pattern units) must have responses less than θ^-

$$\forall l \neq k, \quad 1 \leq i \leq m_l : p_i^l(\mathbf{x}) \leq \theta^- \tag{5}$$

Application of this algorithm to the PNN architecture requires a number of training passes—when the same training dataset is presented to the network repeatedly. This is a significant departure from the PNN architecture in which the dataset is presented only once. This iterative process allows the CPNN to adapt the component densities to achieve optimal coverage of the feature space, and hence increase classification ability in terms of generalisation, sensitivity and specificity.

3. Auscultation data analysis

Data analysis was divided into three parts: spectral computation, where the spectral composition of the tracheal–bronchial sound was established with the Fourier transform; parametric model generation, where a signal estimation model was generated using the Levinson–Durbin algorithm [14]; and a linear normalisation of the resultant estimation model parameters.

Samples were collected for 14 tracheal–bronchial breath sounds as described in Table 1. In order to increase the number of samples available for analysis, further samples were generated from the originals. This was achieved by taking the original sample and varying each point of the signal amplitude by a random amount within the range of $\pm$10%; thereby increasing the global heterogeneity of the dataset by increasing the intra-class variability of the data [15]. These random variations were seeded by Gaussian white noise. For each class of breath sound approximately 20 samples were original whilst the remainder were generated using the aforementioned process.

Table 1 Samples used to train the neural network classifiers, indicating the area of auscultation and the number of samples used in the experiment

Sound	Area of auscultation	Samples
Normal tracheal—bronchial sounds	Tracheal	100
Normal broncho-vesicular sounds	Bronchial	100
Normal vesicular sounds	Bronchial	100
Diminished sounds	Bronchial	100
Tubular breath sounds	Bronchial	100
Fine inspiratory crackles	Bronchial	100
Medium and coarse inspiratory and expiratory crackles	Bronchial	100
Fine, late inspiratory crackles typical of pulmonary fibrosis	Bronchial	100
Mild expiratory wheeze	Bronchial	100
Medium inspiratory crackles with moderate and severe expiratory wheeze	Bronchial	100
Pleural friction rub	Bronchial	100
Inspiratory and expiratory stridor	Tracheal	100
Bone crepitus	Bronchial	100
Subcutaneous emphysema	Bronchial	100

The frequency content of the audio samples was extracted through the use of the fast Fourier transform (FFT) [16]. Rietveld et al. [4] observed a large diversity in frequency spectra between different asthmatic breath sounds, and in conjunction with the results obtained by Folland et al. [15], using frequency spectra as the source of tracheal—bronchial features provided an information-rich representation of the recorded conditions.

Each recorded sample was represented by a *P*-order parametric signal estimation model [14]. This allowed training features to be extracted as the coefficients of the estimation model. In the interests of minimising computational complexity the Levinson—Durbin autoregression algorithm was used to generate the representation. It has been shown that high model estimation uncertainties can arise from spikes in the initial signal [17], and a visual inspection of the data samples revealed no such sharp Fourier spikes that may impede the accuracy of the parametric estimation. The estimated signal was modelled by

$$\hat{s}_x(f) = \frac{\sigma_P^2}{\left|1 + \sum_{k=1}^{P} a_k \, e^{-2j\pi kf}\right|^2} \tag{6}$$

where a_k is the model coefficient and σ_P is the variance. A series of signal estimation models were generated with a varying number of model parameters and inspection of the resultant estimation errors revealed that 15 coefficients would model the signal without an adversely large increase in error. Two constraints could now be imposed on the classification model: the number of input features *I*, and the number of output classes *K*. Using this information in conjunction with the guideline that the number of hidden neurons is of the order $\sqrt{IK}$ [18]—the network capacity could be estimated ergo an approximation as to the size of the training dataset could be established [19]. From this an approximation of 1400 training samples was established. In order to balance the resultant data set, the model parameters were linearly normalised between 0 and 1 to ensure that all parameters contributed equally to the pattern representation when training the ANN classifiers.

4. Data classification

4.1. Preliminary data analysis

Prior to classification by expert examiners and ANN implementations, an assessment of the training dataset was performed. Typically, a classification baseline case would be performed such as linear discriminant analysis (LDA) [8], however low (near-zero) data covariance prevented an accurate appraisal of the linear separability of the available data in its original form. Consequently, another method of dimensionality reduction was applied, to remove the variables that contributed the least information to the dataset. LDA was then able to be performed on the dataset. A typical approach to multivariate data analysis is principal component analysis (PCA) [7,8] that allowed in this instance for a qualitative assessment as to the separability of the dataset and further allowed for the dimensionality of the dataset to be reduced so that LDA could be performed. PCA involves transforming the original *n*-dimensional dataset into *M*-dimensional space by approximating **x**'s *n* constituent orthonormal vectors with an *M* number of vectors. In this instance—to support visual inspection of the

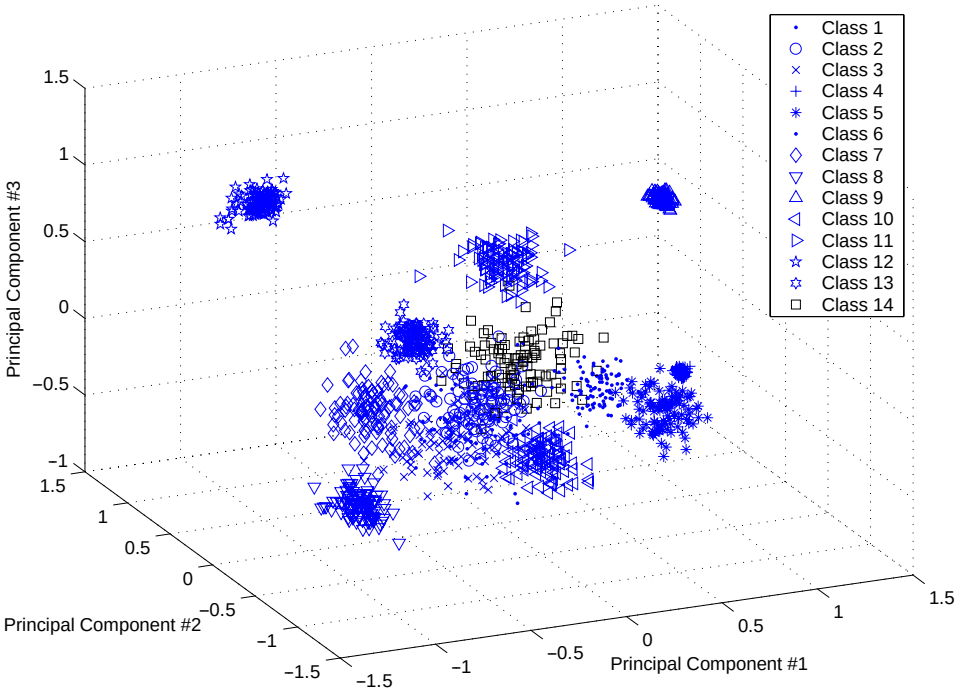

Figure 2 Plot of the first three principal components of the available dataset.

dataset—a three-dimensional representation was selected. Fig. 2 shows the first three principal components of the available training dataset (for clarity, the class labels have been omitted and can be referenced against Table 1). It is observable from Fig. 2 that certain classes are separable in the first three dimensions, notably classes 8, 9 and 12 (fine inspiratory crackles, mild expiratory wheeze, and stridor, respectively) whilst the remainder of the classes cannot be completely linearly separated. Fig. 3 gives the variance captured by the first 10 principal components. From inspection of

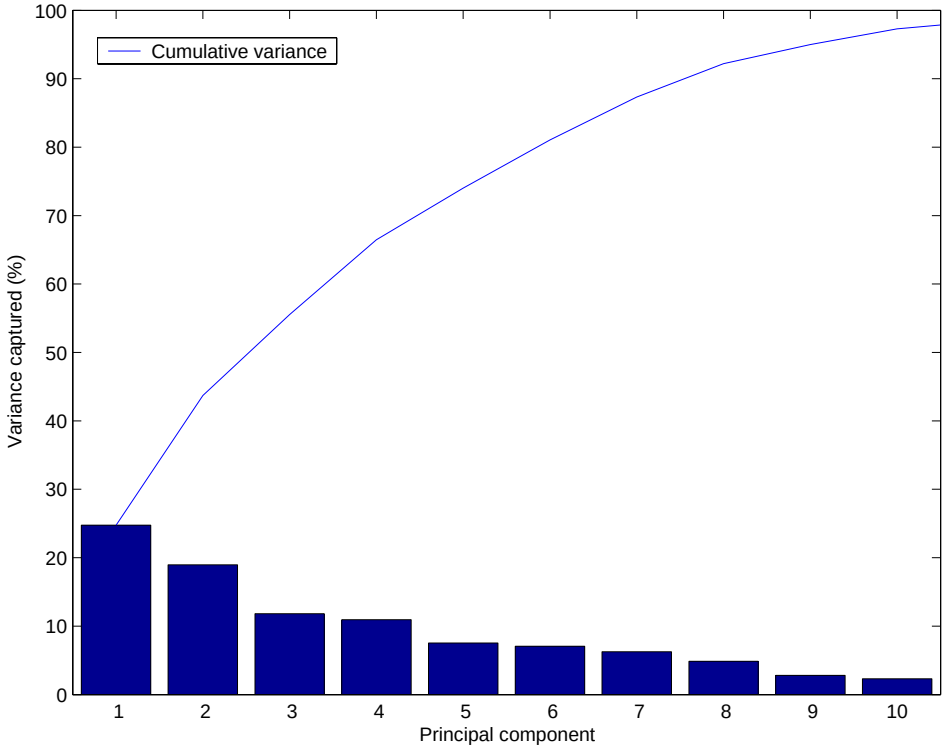

Figure 3 Variance captured by the first 10 principal components.

Fig. 3, the first 7 principal components were kept as these were considered to contribute the most information in the dataset and LDA was then performed across the available data. Four thousand samples from the dataset were retained for the training matrix and 1000 for the sample testing matrix. LDA successfully classified 67.2% of the test dataset.

4.2. Classification by expert human examiners

The data pre-processing stage described in Section 3 resulted in a linearly normalised model coefficient matrix containing one thousand and four hundred 15-point samples. The original samples were presented to four medical examiners: one Senior House Officer (SHO), one Specialist Registrar and two Consultant Anaesthetists. All routinely examine the respiratory system during the daily routine assessment of patients. Each examiner assessed the original tracheal–bronchial sounds and gave a classification based on their knowledge and experience. The accuracy achieved by the expert examiners was 79.3% (specificity 86.6% and sensitivity 54.6%) and a summary of the results is presented in Table 2. It is important to note that the medical examiners were presented with the tracheal–bronchial sounds and were not in contact with the patients. The results obtained from the expert classifications are partially subjective. Whilst the medical examiners were confident in their discrimination between normal tracheal, bronchial and vesicular sounds—reflected by the relatively high classification rate—discrimination between the severities of inspiratory crackles and expiratory wheezes shows less confidence.

4.3. Classification by neural network

Prior to training the CPNN with the dataset, the values of θ^+ and θ^- were defined. In their paper, Berthold and Diamond [12] stated that the respective values of 0.4 and 0.2 provided excellent results in practical situations. It is these values that fundamentally define the sensitivity of the CPNN. The dataset was randomised such that the CPNN was presented with mixed sample patterns. The number of training epochs was set to 5 and the algorithm grew the network entirely within the first training pass. Although no additional neurons were added after the first epoch, adjustments to the pattern units' density functions were made during each training pass. Training was undertaken using the excluding-one-out cross-validation method due to the complexity of the dataset with respect to the number of training patterns available [19]. This method required the training of 1400 individual CPNNs. Each network trained in under 10 min (computer platform: 2×1.67 GHz CPU with 1.5GB RAM) The rate of correct classification was determined as 97.8% (sensitivity 97.8% and specificity 89.6%). Specificity is fundamentally controlled by the region of conflict determined by the initial values of θ^+ and θ^-. Importantly, the number of pattern units created by the CPNN was 390 indicating that the component density functions could be adjusted to incorporate multiple feature space vectors with a high degree of confidence.

The MLP was trained with the same dataset as above. The training algorithm used the Levenberg–

Table 2 Classification results achieved by expert human examiners expressed in class numbers

Sound	Examiner			
	1	2	3	4
1. Normal tracheal–bronchial sounds	(3)	1	1	1
2. Normal broncho-vesicular sounds	2	2	2	2
3. Normal vesicular sounds	(1)	3	(4)	3
4. Diminished sounds	4	4	(6)	4
5. Tubular breath sounds	(7)	5	5	(10)
6. Fine inspiratory crackles	6	6	(8)	(8)
7. Medium and coarse inspiratory and expiratory crackles	(14)	7	(3)	7
8. Fine, late inspiratory crackles typical of pulmonary fibrosis	8	8	(7)	(5)
9. Mild expiratory wheeze	(10)	9	(12)	9
10. Medium inspiratory crackles with moderate and severe expiratory wheeze	10	10	10	(13)
11. Pleural friction rub	(7)	(13)	(13)	11
12. Inspiratory and expiratory stridor	12	12	(10)	12
13. Bone crepitus	13	(14)	(14)	(14)
14. Subcutaneous emphysema	(5)	(7)	(7)	(10)

Incorrect classifications are given in parenthesis.

Marquardt backpropagation, with the MLP employing tangential sigmoid and linear activation functions in the hidden and output layers, respectively. The gradient descent learning algorithm used a momentum term of 0.01. Previous study showed that there were no observable oscillatory effects during learning and hence the momentum term was minimised. Certain algorithms exist that attempt to estimate the topology of feed-forward networks such as the MLP in terms of hidden neurons etc. Examples include the cascade correlation approach [20] and the more recent *genetic* cascade correlation [21]. Here, the number of input and output neurons were fixed to 15 and 14, respectively, whilst the number of hidden layer neurons was increased from the aforementioned $\sqrt{IK}$. From repeated simulations, the optimal number of hidden neurons was determined to be 30. The network was trained for a maximum of 1000 epochs and batch testing of the test data set yielded a classification accuracy of 77.8% under the leaving-one-out cross-validation method. The incorrectly classified patterns spanned all 14 classes equally and as such, demonstrated that there was no particular confusion between classes.

Prior to RBFN training, *K*-means cluster analysis [8] was conducted on the dataset in order to estimate the likely number of radial basis functions required in the RBFN topology. Qualitative assessment of the results revealed that approximately 100 clusters (and hence radial basis neurons) would be appropriate. More clusters did little to improve the accuracy of the *K*-means algorithm as the relationship between the small reduction in distance measure and number of clusters became linear indicating that the algorithm was no longer identifying new clusters. The homoscedastic spread constant σ was set to 0.4. This value was similarly established through a trial-and-error previous study. The training algorithm created a 100-neuron network architecture and validated with the test data yielded a classification accuracy of 96.2%. A comparison between classification accuracies and degrees of freedom is detailed in Table 3.

Table 3 Comparison of the classification accuracies obtained from the medical experts, MLP, RBFN and CPNN neural networks with the respective degrees of freedom

Neural network	Classification accuracy (%)	Degrees of freedom
Experts	79.3	–
MLP	77.8	8
RBFN	96.2	100
CPNN	97.8	390

4.4. Statistical significance testing

Here we firstly define the sensitivity and specificity measures used in the assessment of the ANN classifiers and human experts. From the Bayesian approach, sensitivity can be defined as $p(P|C_k^{\text{true}})$, where P is a positive result (i.e. positive classification) and C_k^{true} is the true class of the breath sound [22]. Similarly we define specificity as $p(N|\bar{C}_k^{\text{true}})$, where N is a negative result (i.e. negative classification). In this case $\bar{C}_k^{\text{true}}$ is used to denote 'not the true class' (for clarity we refrain from referring to $\bar{C}_k^{\text{true}}$ as the *incorrect* or *false* class). Due to 1-in-N output encoding used by the ANNs, 14 individual sensitivity and specificity results were obtained which were then averaged for each ANN.

McNemar's test provides a means by which to assess the statistical significance in difference between the various ANN implementations [23]. Here, the following notation is applied: n_{10} is the number of samples misclassified by ANN B but not by A; and n_{01} is the number of samples misclassified by ANN A but not by B [24]. McNemar's value χ^2 (based on the chi-squared test) can be expressed as

$$\chi^2 = \frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}} \tag{7}$$

where the -1 term is Yates' correction. This term is used as a continuity correction due to the available data being discrete whilst the χ^2 distribution is continuous. In this case, the Null hypothesis was that ANNs A and B have the same error rate. The level of significance was $P = 0.05$ (or 5%), expressed as $\chi^2_{(1,0.95)}$ or 3.84. For the Null hypothesis to have been true (i.e. the difference in error between ANNs A and B was *not* statistically significant) χ^2 would have been below $\chi^2_{(1,0.95)}$. Conversely, for the difference to be considered statistically significant, χ^2 would have been above this value. Table 4 compares

Table 4 Statistical significance of the difference in error rates between the ANN architectures using McNemar's test

Method	LDA (67.2%)	MLP (77.8%)	RBFN (96.2%)	CPNN (97.8%)
LDA	NA	7.37	11.25	11.51
MLP		NA	16.46	14.75
RBFN			NA	7.17
CPNN				NA

NA: not a number.

the ANN error rates using McNemar's test. It can be noted that the results expressed in Table 4 indicate that the difference in error rates between the ANNs are statistically significant, i.e. that the rate of misclassification between the ANNs varies significantly.

5. Discussion

The focus of this paper has been to comparatively assess the performance of the constructive probabilistic neural network against more widely used ANN architectures, specifically the multi layer perceptron and the radial basis function network. The work was conducted in the belief that the complexity of the patterns in feature space (i.e. the distribution of patterns) would impede the MLP and RBFN classification abilities. This is not necessarily the case. The results obtained demonstrate that the MLP and RBFN can discriminate between different classes of tracheal—bronchial breath sounds. However, our attention is drawn to the accuracies obtained by the ANN implementations and to the computational overheads required. In establishing and growing a series of hyper-ellipsoids that transform the feature space to decision space, the CPNN has demonstrated that it can efficiently classify the tracheal—bronchial dataset with a high degree of accuracy and with manageable degrees of freedom.

Using an expert system for the evaluation of the respiratory system is a very attractive concept when standardisation between clinicians appears variable both in this pilot study and in the medical literature [1,2]. For this method to be acceptable it must be at least equal if not better to medical auscultation in terms of sensitivity and specificity [2]. The expert system also allows quantification of a process that until now has been purely qualitative with large inter-observer variability.

6. Conclusions

The results obtained from these ANNs show that the CPNN is more capable of distinguishing between the respiratory sounds produced by disease processes, and is also shown to compare favourably with the results obtained from expert human examiners attaining higher degrees of accuracy, specificity and sensitivity. Further study will be directed towards fusing multiple fuzzy and/or neural architectures to take advantage of their different learning characteristics, in particular to allow the system to linguistically express its decisions in the clinical context. Comparison with a larger group of respiratory medical specialists will also be undertaken. This requirement is highlighted by the results obtained by the medical examiners, who were able to express their degree of confidence in their classifications.

Acknowledgements

The authors would like to thank the medical staff at Birmingham Heartlands Hospital for their assistance in the gathering of data. R. Folland gratefully acknowledges MedicDirect Ltd. for their funding of this research.

References

[1] Bradding P, Cookson JB. The do's and don'ts of examining the respiratory system: a survey of British Thoracic Society members. J R Soc Med 1999;92:632—4.
[2] Wipf JE, Lipsky BA, Hirschmann JV, Boyko EJ, Takasugi J, Peugeot RL et al. Diagnosing pneumonia by physical examination: relevant or relic? Arch Int Med 1999;159:1082—7.
[3] Chedad A, Moshou D, Aerts JM, Van Hirtum A, Ramon H, Berckmans D. Recognition of pig cough based on probabilistic neural networks. J Agric Eng Res 2001;79:449—57.
[4] Rietveld S, Oud M, Dooijes EH. Classification of asthmatic breath sounds: preliminary results of the classifying capacity of human examiners versus artificial neural networks. Comput Biomed Res 1999;32:440—8.
[5] Wilks PAD, English MJ. A system for rapid identification of respiratory abnormalities using a neural network. Med Eng Phys 1995;17:551—5.
[6] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural Networks 1989;2:359—66.
[7] Lin CT, Lee, CSG. Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems. Englewood Cliffs, NJ: Prentice-Hall, 1996.
[8] Bishop C. Neural networks for pattern recognition. Oxford: Clarendon Press, 1995.
[9] Specht D. Probabilistic neural networks. Neural Networks 1990;3:109—18.
[10] Fausett L. Fundamentals of neural networks. Englewood Cliffs, NJ: Prentice-Hall, 1994.
[11] Funahashi K. Multilayer neural networks and Bayes decision theory. Neural Networks 1998;11:209—13.
[12] Berthold M, Diamond J. Constructive training of probabilistic neural networks. Neurocomputing 1998;19: 167—83.
[13] Streit RL, Luginbuhl TE. Maximum likelihood training of probabilistic neural networks. IEEE Trans Neural Networks 1994;5:764—83.
[14] Luo F, Unbeahauen P. Applied neural networks for signal processing. Cambridge: Cambridge University Press, 1997.
[15] Folland R, Hines EL, Boilot P, Morgan DW. Classifying coronary dysfunction using neural networks through cardiovascular auscultation. Med Biol Eng Comput 2002; 40:339—43.
[16] Strum R, Kirk D. Contemporary linear systems using MATLAB 4.0. Boston: PWS, 1996.

[17] Christini D, Kulkarni A, Rao S, Stutman E, Bennett F, Hausdorff J et al. Influence of autoregressive model parameter uncertainty on spectral estimates of heart-rate dynamics. Ann Biomed Eng 1995;23:127—34.
[18] Widrow B, Lehr MA. 30 years of adaptive neural networks: perceptron, madaline and backpropagation. Proc IEEE 1990;78:1415—42.
[19] Tarrasenko L. A guide to neural computing applications. London: Arnold, 1998.
[20] Fahlman SE, Lebiere C. The cascade correlation learning architecture. Technical Report, CMU-CS-90-100. School of Computer Science, Carnegie Mellon University, 1990.
[21] Liang H, Dai G. Improvement of cascade correlation learning algorithm with an evolutionary initialization. Inform Sci 1998;112:1—6.
[22] Congdon P. Bayesian statistical modelling. Chichester: Wiley, 2001.
[23] Campbell MJ, Machin D. Medical statistics: a commonsense approach. Chichester: Wiley, 1999.
[24] Roggo Y, Duponchel L, Huvenne J-P. Comparison of supervised pattern recognition methods with McNemar's statistical test: application to qualitative analysis of sugar beet by near-infrared spectroscopy. Anal Chim Acta 2003;477:187—200.