NOVELTY, CONFIDENCE & ERRORS IN CONNECTIONIST SYSTEMS

Stephen J. Roberts [1], William Penny [1] & David Pillot [2]

Introduction

Artificial 'neural' networks (ANNs) have become very popular for data analysis over the past decade. In particular, feedforward 'neural' networks have, it may be argued, become so popular because they can, as classifiers, estimate *a posteriori* probabilities *directly* by forming a mapping function from the data space to a probability space [3] and as regressors may correctly estimate the conditional average of target data on a given input.

If, however, we are to exploit the undoubted utility of ANNs in *safety critical* environments then classification or regression performance in itself is not enough. One of the key requirements of any statistical analysis system should be to assess its own confidence in a decision in the case of classification, and estimate probable error bars in the case of regression. Part of the problem for any Bayesian classifier is the fact that the posteriors, by definition, sum to unity. This means that a classification is made into one of a closed set of classes. If 'rogue' data appears then, even if it fails to conform to the statistics of 'genuine' data, it will be classified *with apparent confidence* into one of the output classes. We must, therefore, monitor the confidence in any classification decision [4, 5]. This paper looks at some of the issues in estimating errors and confidence limits in feed-forward networks, and results are presented on a simple regression problem and an example from medical diagnostics.

Error and Uncertainty

We define firstly a training data set, $\mathcal{D}$, which consists of N examples of input-output pairs, $\mathbf{x}^n; \mathbf{t}^n$, where $\mathbf{x}^n$ is the n-th input (feature) vector and $\mathbf{t}^n$ is the target vector associated with it[4]. We use a feedforward network to estimate a mapping function conditional on $\mathcal{D}$, $\mathcal{F}_{\mathcal{D}}$ say, from input to output space such that $\mathcal{F}_{\mathcal{D}} : \mathbf{x} \rightarrow \mathbf{y}$ where $\mathbf{y}$ is the output vector of the network.

Unless we have noiseless and separable data (i.e. there is no intrinsic Bayes' error) then certain regions of input space have a higher mapping error (when assessed against the set of $\mathbf{t}^n$) associated with them. We may attempt to estimate this error by training another mapping function, $\mathcal{G}_{\mathcal{D},\mathcal{F}}$, conditional on both $\mathcal{D}, \mathcal{F}$ such that $\mathcal{G}_{\mathcal{D},\mathcal{F}} : \mathbf{x} \rightarrow \boldsymbol{\epsilon}$, such that $\boldsymbol{\epsilon}^n$ is the vector of square errors between each output and its target i.e. $\boldsymbol{\epsilon} = \text{diag } (\mathbf{y}^n - \mathbf{t}^n)(\mathbf{y}^n - \mathbf{t}^n)^T$. Like $\mathcal{F}$, however, $\mathcal{G}$ only provides reliable estimates for the errors when the input is in the finite region of space, $R(\mathcal{D})$, spanned by $\mathcal{D}$. For an *unseen* input $\mathbf{u}$ both $\mathbf{y}, \boldsymbol{\epsilon}$ are meaningless if $\mathbf{u} \notin R(\mathcal{D})$. For data that is *novel* with respect to $R(\mathcal{D})$ this means that the network's outputs are not valid and errors are very large (confidence close to zero). Previous attempts have been made to devise confidence measures for neural networks and these may be broken into three basic methods; Generating an artificial 'rogue' data class outside $R(\mathcal{D})$ [5], using non-parametric density estimators [1, 4] and using Bayesian-inspired techniques such as marginalisation [3, 2]. Although the artificial class method works well on low-dimensional problems, it scales poorly and is computationally costly; the density estimation techniques presuppose that regions of low data density generate the largest errors and this is not the case in many data sets [4]; Bayesian techniques, if implemented fully, are computationally very costly, current methods of approximation [3, 2] are often unrealistic when dealing with complex data sets and also fail to give a valid result in the classification case.[5] We present here a method

[1]Department of Electrical & Electronic Engineering, Imperial College, University of London.

[2]Ecole Polytechnique de Compiegne, France

[3]Thus bypassing the problematical estimation of class-conditional likelihoods which suffer notoriously from the 'curse of dimensionality'.

[4]This will be 'one-of-n' coded if we are dealing with a classification problem.

[5]Uncertainty or errors escalate away from the data set but only along the path of class decision boundaries.

which offers an effective means of estimating network errors and confidence values. The method relies upon the use of committees of networks and the theory behind this is dealt with in the next section.

Theory

Any network mapping function may be parametrically represented by a weights vector, $\boldsymbol{\theta}$. The use, however, of a single weights vector (as is common practise) is sub-optimal unless that vector dominates the posterior weights space. As most problems are complex, this scenario is highly unlikely (i.e. there are many energy minima). Although we can optimise $\boldsymbol{\theta}$ this vector merely represents the weights set corresponding to one of many minima in the network's energy function (or peak in the posterior distibution of $\boldsymbol{\theta}$). A Bayesian approach provides a paradigm shift such that efforts in parameter optimisation of a *single* weights vector may be replaced by efforts in integrating over *many* competing weights vectors. A full solution is thus, in theory, achieved by integrating out the 'nuisance vector', $\boldsymbol{\theta}$; this is referred to as *marginalising.*

$$\mathbf{y}(\mathbf{u}) = \int_{\text{all } \boldsymbol{\theta}} \mathbf{y}(\mathbf{u} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}) d\boldsymbol{\theta} \tag{1}$$

This integral may be approximated by only integrating $\boldsymbol{\theta}$ over small regions, r_i around each of the many non-equivalent minima, m_i, of the energy function.

$$\mathbf{y}(\mathbf{u}) = \sum_{m_i} p(m_i \mid \mathcal{D}) \int_{r_i} \mathbf{y}(\mathbf{u} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}, m_i) d\boldsymbol{\theta} \tag{2}$$

In this paper we take a simplified approach which removes the computationally intensive operation of numerically integrating around each m_i by assuming that $p(\boldsymbol{\theta} \mid \mathcal{D}, m_i)$ is sharply peaked with peak value α_i', hence

$$\mathbf{y}(\mathbf{u}) = \sum_{m_i} p(m_i \mid \mathcal{D}) \alpha_i' \mathbf{y}(\mathbf{u} \mid \boldsymbol{\theta}_i) \tag{3}$$

Letting $p(m_i \mid \mathcal{D})\alpha_i' = \alpha_i$ and $\mathbf{y}(\mathbf{u} \mid \boldsymbol{\theta}_i) = \mathbf{y}_i$ so Equation (3) reduces to

$$\mathbf{y}(\mathbf{u}) = \sum_i \alpha_i \mathbf{y}_i \tag{4}$$

i.e. a *committee* of networks, each generated by, for example, starting a training process with a different set of random number seeds. The values of α_i may be generated from the committee error covariance matrix [2] or simply set to $1/M$ so that the committee becomes an unweighted averaging over M networks (this is the approach taken in this paper).

We consider a square error contribution of the standard form for the k-th component of the output vector, $E_k = (y_k - t_k)^2$ where t_k is the appropriate target vector for y_k. Taking expectations over members of the committee and expanding the square gives

$$\mathcal{E}[E] = \mathcal{E}[y_k^2] + t_k^2 - 2t_k\mathcal{E}[y_k]$$

given that the variance of y_k over the committee is

$$var[y_k] = \mathcal{E}[y_k^2] - \mathcal{E}[y_k]^2$$

so we may write the expected error as

$$\mathcal{E}[E_k] = \overline{E}_k = var[y_k] + (\mathcal{E}[y_k] - t_k)^2 \tag{5}$$

This is an intuitively pleasing result as the error may be regarded in a common-sense manner as arising from two distinct causes. The first term penalises variant decisions *between* committee members and the second penalises the committee *as a whole* if the output is erroneous. Both terms are easily calculated, the second being given, from Equation (4) and our definition of ϵ_k, as

$$(\mathcal{E}[y_k] - t_k)^2 = \sum_{i \in committee} \alpha_i \epsilon_{i,k} \tag{6}$$

We have, however, neglected some other (smaller) contributions to the overall error in the interests of space (a forthcoming paper will cover these issues in detail). Firstly, in the derivation of Equation (3) from (2) we assumed that each $p(\boldsymbol{\theta} \mid \mathcal{D}_i, m_i)$ was sharply peaked. We note that the error associated with this approximation is directly related to the Hessian matrix at each error minimum (each member of the committee). Secondly, in the same way that we gain a contribution to the error by evaluating expectations over an ensemble of possible networks (the committee), so another error contribution exists as we are training using a *single realisation* of $\mathcal{D}$ and ideally we would like to observe the variance over an ensemble of possible data realisations.

If wish to use $\overline{\mathbf{E}}$ to provide information about data outside $R(\mathcal{D})$, we must use a network such that for changes in weights vectors between networks, $var[\mathbf{y}]$ *increases* with distance away from $R(\mathcal{D})$. For a multi-layer perceptron (MLP) this is true, but for a radial-basis function (RBF) network with Gaussian kernels this is not the case. For the RBF, the output may be given as a linear combination of hidden-layer responses, $\boldsymbol{\Phi}$ plus a bias term, $\boldsymbol{\beta}$

$$\mathbf{y} = \mathbf{W}\boldsymbol{\Phi} + \boldsymbol{\beta} \tag{7}$$

where $\mathbf{W}$ is the weights matrix of the linear final layer. As the bias term compensates for differences in the means of the set of output vectors and the set of target vectors, i.e.

$$\boldsymbol{\beta} = \langle \mathbf{t} \rangle - \langle \mathbf{y} \rangle$$

it is simple to show that for *any* committee of RBF networks, there is little difference between the bias terms of the members. If we require, therefore, the variance of Equation (7) to rise away from $R(\mathcal{D})$ we must choose the hidden-layer function such that its response *rises* with distance from its location in input space. This therefore precludes the use of Gaussians. We instead utilise another universally approximating function, the *thin-plate spline* (TPS).

All the results in this paper, unless otherwise stated, are created from a committee of RBF networks each utilising thin-plate spline functions on the hidden layer, such that, if $r = |\mathbf{x} - \boldsymbol{\mu}|$ where $\boldsymbol{\mu}$ is the location of the spline kernel

$$\phi(r) = r^2 \ln(r)$$

The final-layer weights matrix is thence easily calculated via matrix psuedo-inversion. The locations of the splines were randomly located within $R(\mathcal{D})$ for each network. This gives rise to networks which are extremely fast to generate (taking less than two seconds of CPU time on an Ultra-SPARC to generate the entire committee for the problems in this paper). The number of basis functions was not optimised for the examples given, however, as this is a separate issue which we will not deal with here.

<u>Confidence</u>

As detailed, we may estimate the square error associated with each network output. For the case of a regression problem this provides input-dependent error bars. We would like, however, to obtain some measure of decision confidence for classification problems. For optimal classification we choose the class with the largest network output (this being the largest posterior), y_m say. We allow this output, however, to be defined, not as a single number, but in a set-based Bayesian manner as a subinterval in $[0, 1]$. For an error estimate $\sigma_k = \sqrt{\overline{E}_k}$ so this subinterval spans $[y_k - \sigma_k, y_k + \sigma_k]$. We thence define a simple measure of confidence as the proportion of $[0, 1]$ *not* contained in this subinterval i.e.

$$c_k = 1 - 2\sigma_k \tag{8}$$

Applications & Results

We first show an example regression problem. The system has been trained on noisy sine function which has been defined only in the x (input) interval $[100, 200]$. Subsequent presentation of inputs over a larger range gives rise to the regression and error results shown in Figure (1). The regressor consists of a committee of ten TPS networks, each with five basis functions. The errors bars are plotted at $\pm 2\sigma$. Note the rise in error as extrapolation occurs.

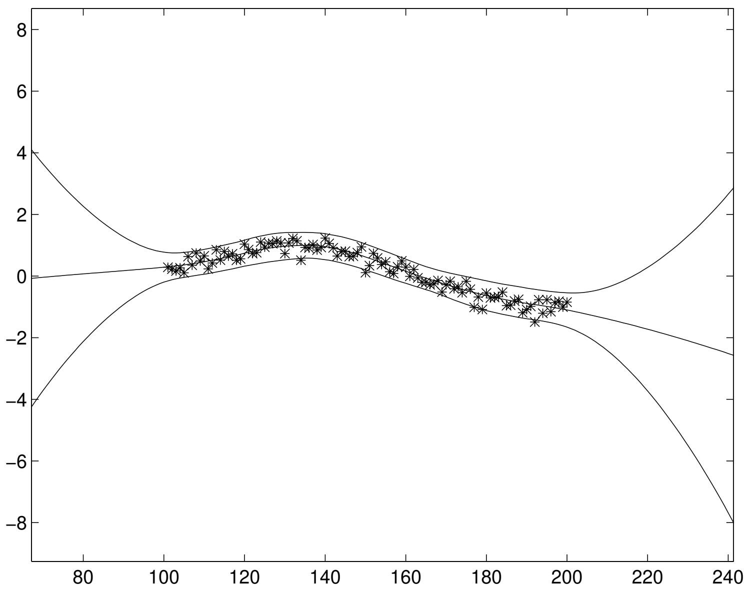

Figure 1: Regression function and input-dependent error bars.

We show next an application of the techniques developed on a real problem of muscle-tremor classification. The aim of the classification was to distinguish between patient and normal groups. The patient group consisted of nine patients with (primarily) Parkinson's disease or multiple sclerosis. Arm muscle tremor was recorded using a 3-D mouse with a magnetic pickup which tracked movement in three linear and three angular directions. The time series from the measurements were parameterised using a set of autoregressive (AR) models. The number of features is reduced to two for the purposes of this paper as it allows easy visualisation. Figure (2) shows the training set (99 vectors of each class). Class 0 represents non-patients and class 1 patients. We note that data from class 1 (patients) forms neatly separated regions for the most part, but that several data points fall within the same region as non-patient data. Without measurement of confidence we run the risk of falsely classifying patients as non-patients or *vice versa*.

Figure (3) shows the *test* set superimposed on a grey-scale image of confidence. The plot is over a region of space 1.5 times larger than that spanned by the training set. We see that confidence drops rapidly outside $R(\mathcal{D})$, but that within the region of the data, where $var[\mathbf{y}]$ is low, the second term of Equation (5) provides an indication of regions of low network confidence. We note that the region of overlapping data is of lower confidence.

It is common practise in statistical decision making to classify data to a 'rogue' data class if some test statistic (commonly the maximum class posterior) is less than some threshold. If we allow the threshold to vary and evaluate the error rate and classified fraction of the data set then we obtain operating characteristics for the classifier; the aim being, of course, to reduce the error rate as far as possible whilst classifying as large a fraction of the data as possible. Figure (4) shows the operating curve of error rate against classified fraction using a sliding threshold based on confidence and a committee classifier (upper line) and a single, ten-spline network using a threshold on the maximum class posterior (lower line). We note that the former offers dramatically better

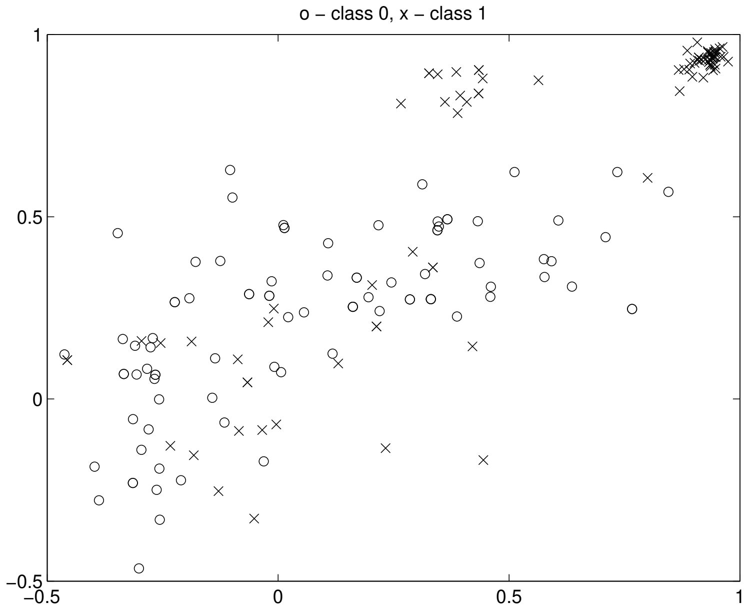

Figure 2: Training data set.

performance. It is possible, for example, to classify 80% of the data set with an accuracy of 90% using the committee-confidence approach, whilst the single network can only achieve the same accuracy on 20% of the data set.

Conclusions

It is clear that, if we are to apply connectionist techniques to any data, some clear understanding of how erroneous network outputs are is required. We have presented results and theory in this paper that provide an effective method for providing such an understanding. Furthermore, a committee of networks is guaranteed to outperform the mean performance of a set of individual networks [2].

Acknowledgements

The authors would like to thank Dr. Maria Stokes and Julia Spyers-Ashby for the tremor data and many helpful discussions. WP is a research fellow funded by EPSRC (GR/K79062), whose support we gratefully acknowledge. DP was a visiting researcher at IC during summer 1996.

References

[1] C.M. Bishop. Novelty Detection and Neural Network Validation. IEE Proc. Vis. Image Signal Processing, 141:217–222, 1994.

[2] C.M. Bishop. Neural Networks for Pattern Recognition. Oxford University Press, Oxford, 1995.

[3] D.J.C. MacKay. A Practical Bayesian Framework for Backpropagation Networks. Neural Computation, 4:448–472, 1992.

[4] S. Roberts and L. Tarassenko. A Probabilistic Resource Allocating Network for Novelty Detection. Neural Computation, 6:270–284, 1994.

[5] S. Roberts, L. Tarassenko, J. Pardey, and D. Siegwart. A Validation Index for Artificial Neural Networks. Proceedings of International Conference on Neural Networks & Expert Systems in Medicine & Healthcare, Plymouth, August 1994, pages 23–30, 1994.

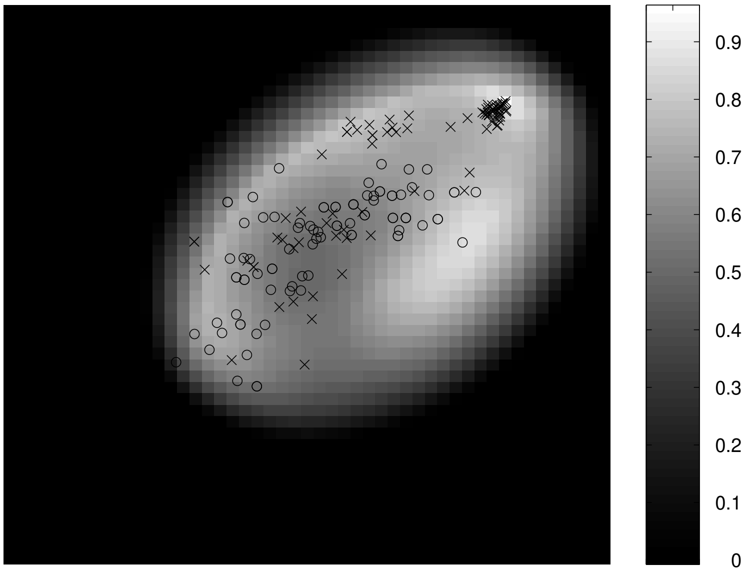

Figure 3: Grey-scale coding of confidence measure superimposed with test data set.

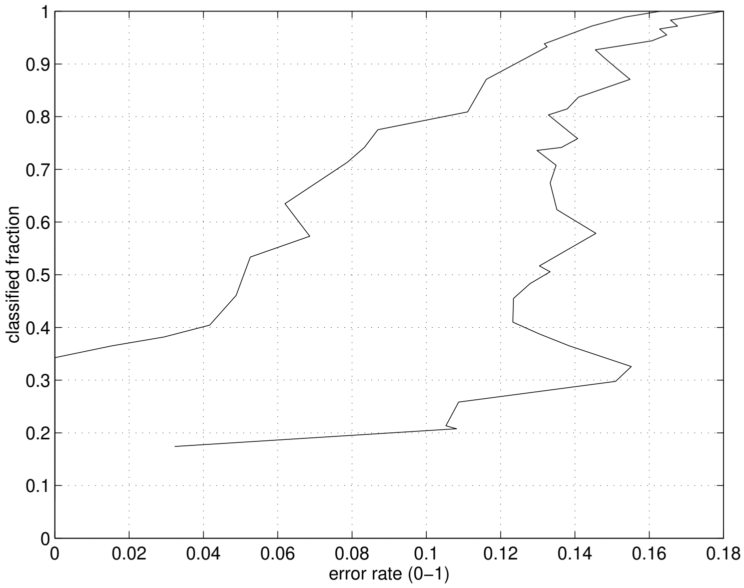

Figure 4: Operating curves for the committee-confidence approach (upper) and a single TPS-RBF network using a maximum class posterior rejection threshold (lower).