A Comparison of Feature Extraction and Selection Techniques

J F Dale Addison, Stefan Wermter, Garen Z Arevian

Abstract. We have applied several dimensionality reduction techniques to data modelling using neural network architectures for classification using a number of data sets. The reduction methods considered include both linear and non linear forms of principal components analysis, genetic algorithms and sensitivity analysis. The results of each were used as inputs to several types of neural network architecture, specifically the performance of Multi-layer perceptrons, (MLPs), Radial basis function networks (RBFs) and Generalised regression neural networks. Our results suggest considerable improvements in accuracy can be achieved by the use of simple network sensitivity analysis, compared to genetic algorithms, and both forms of principal component analysis.

Index Terms-Dimensionality reduction, feature extraction, feature selection, neural networks

I. INTRODUCTION

In this work we have considered a number of means of improving the classification accuracy of neural network models by reducing the dimensionality of the data set. There is a trade-off between accuracy as represented by the entire data set and the computational overheads of retaining all parameters without application of feature extraction/selection techniques. This is referred to as the "curse of dimensionality" [1]. Our work considers the merits of feature extraction where the original variables are retained but processed into a smaller set to retain as much information as possible, and feature selection which removes input variables that do not contribute significantly to model performance [13,10,2].

We have applied several different techniques to this problem, specifically principal components analysis (linear and non-linear), sensitivity analysis and genetic algorithms to seven data sets which differ in attribute and feature size and whose complexity range from stable distinctive class structures to highly overlapping class structures. We have structured the paper accordingly. Section 2 discusses the feature extraction and selection techniques used. Section 3 discusses neural network modelling techniques from the perspective of hyper planes and hyperspheres. Section 4 outlines the method used and tabulates the results of our experiments. Section 5 discusses the results and offers some conclusions as to the effectiveness dimensionality reduction in relation to neural networks.

II. DIMENSIONALITY REDUCTION TECHNIQUES

By reducing the dimensionality of the input set correlated information is eliminated at the cost of a loss of accuracy [10]. Dimensionality reduction can be achieved either by eliminating data closely related with other data in the set, or combining data to make a smaller set of features. The feature extraction techniques used in this study are principal components analysis and autoassociative neural networks Feature selection is achieved by the use of genetic algorithms, sensitivity analysis.

A. Linear Principal Components Analysis

Using the example of projecting data from two dimensions to one, a linear projection requires the optimum choice of projection to be a minimisation of the sum-of-squares error [3, 4]. This is obtained first by subtracting the mean $\overline{x}$ of the data set. The covariance matrix is calculated and its eigenvectors and eigenvalues are found. The eigenvectors corresponding to the *M largest* eigenvalues are retained, and the input vectors x^n are subsequently projected onto the eigenvectors to give components of the transformed vectors z^n in the *M*-dimensional space. Retaining a subset $M < d$ of the basis vectors μ_i so that only M coefficients z_i are used allows for replacement of the remaining coefficients by constants b_i. This allows each x vector to be approximated by an expression of the form:

$$\tilde{\chi} = \sum_{i=1}^{M} z_i u_i + \sum_{l=M+1}^{d} b_i u_i \qquad (1)$$

Where μ_i represents a linear combination of *d* orthonormal vectors.

B. Auto Associative Networks (AAN)

An auto associative network (AAN) consists of a multi-layer perceptron with *d* inputs, *d* outputs, and *M* hidden units with $M < d$. [5]. The targets used to train the network are the input vectors themselves, which means the network is attempting to map each input vector onto itself. Because the number of units in the middle layer is reduced, a perfect reconstruction of the input vectors may not always be possible. The network is

then trained using a sum of squares error of the following form.

$$E = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{d}\{y_k(x^n) - x_k^n\}^2. \quad (2)$$

where N is the number of patterns in the sample and x_k^n represents the target value for the output unit k when the input vector is x^n. The error minimisation here performs a form of unsupervised training, even though we are using a supervised architecture as no independent target data is provided. Such networks perform a non-linear principal components analysis, which has the advantage of not using linear transformations.

C. Genetic Algorithms

We use the Holland [6] algorithm. The algorithm can be expressed as follows.

(1) Set generation counter I ← 0
(2) Create initial population, Pop(i), by random generation of N-individuals
(3) Apply objective function to the individual, record the value found (determines data fitness)
(4) Increment next generation, I ←I +1
(5) Select N individuals randomly from the previous population Pop (I-1) based on their fitness
(6) Select R parents from new population to form new population to form new children by applying the genetic operators
(7) Evaluate fitness of newly formed children by applying the objective function
(8) If I < maximum number of generations to be considered, go to step (4)
(9) Write out best solution found

D. Sensitivity Analysis

We conducted sensitivity analysis by treating each input variable in turn as if it were "unavailable" [13]. A neural network is trained using all of the input attributes, and the values of training and test set errors are produced. Afterwards the network is "pruned" of input variables whose training and verification errors are below the threshold. In this way variables can be assessed according to the deterioration effect they have upon network performance if removed.

III. Architectures

We now consider neural network architectures and training algorithms for classification purposes. Our study has compared and contrasted neural networks and several different training algorithms. Multi-layer perceptrons and radial basis function networks represent two different aspects of non-linear function mapping. The former computes a non-linear function of the scalar product of the input and weight vector, according to the following equation:

$$y_k = \tilde{g}\left(\sum_{j=0}^{M} w_{kj}^{(2)} g\left(\sum_{i=0}^{d} w_{ji}^{(1)} x_i\right)\right) \quad (3)$$

where y_k is the activation of the kth output unit and $\tilde{g}$ is a function which absorbs the bias of the network into its weights in the linear combination of the outputs of the hidden units, and g is a weighted linear combination of the d input values. By contrast Radial basis function networks determine the activation of a hidden unit by measuring the distance between the input and prototype vector. The posterior probability of class membership is given by

$$P(C_k \mid x) = \sum_{j=1}^{M} w_{kj}\theta_j(x) \quad (4)$$

Where j represents an index for a common pool of M basis functions for each of the class conditional densities and the second layer weights w_{kj} represent the probability of a class being a member of a particular density. Multi-layer perceptrons [7] were trained using several different training algorithms. Back propagation [8] demonstrates how the derivatives of the error function with respect to the network parameters can be obtained in a computationally efficient way. Indeed the use of gradient information is of central importance in algorithms for network training. These algorithms are a well-established branch of techniques to minimise continuous differentiable functions of many different variables. The simplest of these are gradient descent techniques. The batch version of which is given by the equation

$$\Delta w^{(\tau)} = -\eta \nabla E^n\Big|_{w^{(\tau)}} \quad (5)$$

where $w^{(0)}$ represents an initial guess as to the weight vector, η represents the learning rate, τ the distance moved in the direction of the greatest rate of decrease of the error (in the direction of the negative gradient) evaluated at $w^{(\tau)}$.

IV. Methods and Results

Table 1 gives further details of the 6 data sets used in this studyts to test both the reduction methods, and the neural network architectures.. Tables 2-6, show the performance of each technique used. A benchmark of performance was established by training each architecture on the full data set, without the benefit of dimensionality reduction (referred to in each table as "Normal"). Then each method was applied in turn. The Abalone, Horse Colic, and Pima Indians, data sets can be found at the UCL database repository [14], whilst the machine tool and elevator sets are from the MINICON project from the University of Sunderland. In each case the data was partitioned into training, verification and test sets in the ratio 70-20-10 respectively. The results shown are for the verification set.

Table 1: Details of Data sets used

Data set	Attribs	Inst	Class	Comment
Elevator	7	14000	2	Large data set. Some noise in one parameter which increases the dimensionality
Abalone	8	4177	3	Highly overlapping classes, highly unstructured domain
Horse colic	27	368	2	Mix of continuous, discrete and nominal. 30% of attributes missing.
Pima Indians	8	768	2	High degree of Kurtosis, highly overlapping classes
Sonar	60	208	2	Mix of nominal and discrete attributes
Machine tool	401	141	2	Well structured, non overlapping class structure

Table 2: Results of elevator data set

		RBF	MLP	PNN
Technique	**Inputs**	**Result**	**Result**	**Result**
Normal	8	81%	84%	77%
PCA	5	74%	71%	75%
AAN	4	74%	78%	75%
Gen Alg	5	78%	78%	77%
Sen Anal	2	76%	77%	74%

Table 3: Results of Abalone data set

		RBF	MLP	PNN
Technique	**Inputs**	**Result**	**Result**	**Result**
Normal	8	91%	92%	58%
PCA	4	91%	92%	58%
AAN	2	82%	82%	32%
Gen Alg	1	86%	87%	34%
Sen Anal	3	90%	91%	90%

Table 4: Results of Horse colic data set

		RBF	MLP	PNN
Technique	**Inputs**	**Result**	**Result**	**Result**
Normal	27	90%	88%	90%
PCA	6	85%	90%	83%
AAN	3	68%	60%	68%
Gen Alg	21	90%	90%	90%
Sen Anal	3	90%	92%	90%

Table 5: Results of Pima Indians

		RBF	MLP	PNN
Technique	**Inputs**	**Result**	**Result**	**Result**
Normal	8	62%	68%	70%
PCA	3	82%	81%	80%
AAN	4	60%	66%	69%
Gen Alg	7	67%	68%	79%
Sen Anal	5	76%	69%	72%

Table 6: Results of Sonar data set

		RBF	MLP	PNN
Technique	**Inputs**	**Result**	**Result**	**Result**
Normal	60	98%	95%	88%
PCA	10	73%	75%	63%
AAN	3	50%	82%	48%
Gen Alg	47	90%	98%	78%
Sen Anal	47	95%	98%	78%

Table 7: Results of machine tool data set

		RBF	MLP	PNN
Technique	**Inputs**	**Result**	**Result**	**Result**
Normal	401	100%	100%	85%
PCA	6	100%	100%	95%
AAN	200	75%	100%	80%
Gen Alg	148	100%	90%	90%
Sen Anal	9	100%	100%	84%

V. DISCUSSIONS AND CONCLUSIONS

This study has evaluated the effectiveness of feature extraction and selection techniques applied to data modelling using neural networks. The most noticeable effect is their reduction in accuracy upon the probabilistic neural networks. Two data sets in particular, machine tool and sonar, which have high dimensionality feature spaces show degradation in classification ability when features are removed. This can be explained by the nature of Probabilistic networks which require storage of all data points which it uses to initiate the kernel function approximators. Removal of substantial numbers of features prior to training will interfere with the assignment of cases to the kernel based estimators and thereby reducing classification accuracy. This is also noted in the results for the radial basis function networks, although the effect is far less severe than when the outputs from the autoassociative networks are used. The most effective techniques in this study were found to be principal components analysis and sensitivity analysis. In four out of the seven sets used they were superior to other methods

considered. This is not so surprising in the case of sensitivity analysis which is able to replace attributes ad infinitum until a suitable model is achieved, and is not constrained by the stopping criteria used with networks where the dimensionality reduction technique was used "off line". This needs to be balanced against the length of time required to compare and contrast the viability of the features retained for modelling. The results for principal components analysis suggests that even data sets whose class structure is highly overlapping contain enough information to allow accurate variance calculations for principal components determination.

REFERENCES

[1] Bishop, C. M 1997 "Neural networks for pattern recognition," Oxford University Press, pp 6-9

[2] Addison, J F D, Wermter, S and MacIntyre, J. 1999 *Effectiveness of feature extraction in neural network architectures for novelty detection,* ICANN-99, Ninth International Conference on Artificial Neural Networks", Edinburgh, UK, September 1999, pp976-981.

[3] Bishop, C. M 1995 "Neural networks for pattern recognition", Oxford University Press, pp 310-313

[4] Press, W.H, S. A. Teukolsky, W.T Vetterling, and B.P Flannery 1992 "Numerical recipes in C: The art of scientific computing "(second edition), Cambridge University press

[5] Fausett, L. 1994, "Fundamentals of Neural Networks," Englewood Cliffs, NJ: Prentice Hall.

[6] Holland, J.; 1975. "Adaptation in Natural and Artificial systems. MIT Press".

[7] Haykin, S.; 1995 "Neural Networks - A Comprehensive Foundation.;" Maxwell Macmillan International Publishing Company, pp 138

[8] Rumelhart, D. E.; Hinton, G. E.; and Williams, R J. 1986. Learning Internal Representations by Error Propagation in Parallel Distributed Processing: Explorations in the *Microstructure of Cognition:* 318-362

[9] Hestenes, M. R. and Stiefel, E. 1952 "Methods of Conjugate Gradients for Solving Linear Systems". *Journal of Research of the National Bureau of Standards* 49(6): 409-436.

[10] Addison, J F D, MacIntyre J (editors) 2003 "Intelligent techniques: A review", Springer Verlag (UK) Publishing Company, 1st Edition, Chapter 9, *To appear September 2003*

[11] Lowe, D.; 1995 "Radial Basis Function Networks, The Handbook of Brain Theory and Neural Networks", Cambridge, MA: MIT Press

[12] Specht, D. F.; 1990 "Probabilistic Neural Networks". *Neural Networks* 3 (1): 109-118.

[13] Hunter, A., Kennedy, L., Henry, J. and Ferguson, R.I. 2000, "Application of Neural Networks and Sensitivity Analysis to improved prediction of Trauma Survival" *Computer Methods and Algorithms in Biomedicine* 62, pp. 11-19

[14] Blake, C.L & Merz, C.J. http:/www.ics.uci.edu/~mlearn/ML Repository

[15] Roelof, R K., Pedrycz, W.; "One to many mappings represented on feed forward network"*s*, *ESANN 2001 proceedings* – European symposium on Artificial neural networks, Bruges, Belgium, 25-27 April 2001, p365-370

[16] Wermter S, Sun, R,; 2000, "Hybrid Neural Systems," Springer-Verlag Publishing Company, Heidelberg, Germany