Bidirectional Neural Networks Reduce Generalisation Error

A.F. Nejad[1] and **T.D. Gedeon**[1,2]

[1] School of Computer Science and Engineering
The University of New South Wales
Sydney, 2052, AUSTRALIA

[2] Department of Telecommunications and Telematics
The Technical University of Budapest
Sztocek u. 2, H-1111, HUNGARY

Abstract:

BiDirectional Neural Networks (BDNN) are based on Multi Layer Perceptrons trained by the error back-propagation algorithm. They can be used as both associative memories and to find the centres of clusters.

One of the major challenges in neural network research is data representation. We have used cluster centroids obtained by BDNNs and some heuristic techniques to achieve good representations. This is the key factor in reducing generalisation error. Evaluation of these methods is done by statistical learning theory supported by experimental results. A variety of data sets from real-world problems have been used to support the results of our methods.

The results are consistent with the Vapnik-Chervonenkis bounds. Our methods can be considered as efficient means of designing the required bias in solving dynamic and complex learning systems and to increase their expected performance.

1. Introduction

Neural networks, in particular multi layer perceptrons trained by the error back-propagation learning algorithm (Rumelhart et al, 1986), have been studied extensively by researchers. This learning method can be considered as a type of non parametric model-free estimator.

Two of the major recent challenges for researchers in neural networks have been improving learning models, and data representation. For the former, we designed BiDirectional neural networks (Nejad and Gedeon, 1994), which are powerful models based on feed-forward neural networks. We have shown that BDNNs are effective tools for classification, prediction, associative representation, and finding the centroid of a cluster.

The data representation problem is concerned with appropriate form of input or output transformation. A good representation is often more important than the learning algorithm and does most of work. This is emphasised by many researchers (eg. Anderson, 1988; Stuart, 1992). However this has been studied much less than learning models.

This paper is concerned with data representation techniques to reduce generalisation error and balance bias and variance of the learning algorithms. The experiments have been done both the standard

error back-propagation algorithm and BiDirectional Neural Networks on four real-world data sets and some artificial data sets. Vapnik-Chervonenkis (VC) dimension is used to study the consistency between average generalisation performance and the worst case bounds obtained from formal learning theory (Blumer, 1989; Haussler, 1990). The representational capacity of BP and BDNN are measured by the VC dimension and support the results of our experiments by the suggested methods. We show that BDNNs are appropriate tools for enhancing data representation techniques and partitioning effectively the feature space and reducing generalisation error.

The organisation of this paper is as follows. Section 2 introduces bidirectional neural networks. A brief description of the data sets used in our experiments will appear in section 3. Section 4 very briefly reviews the data representation problem and popular methods of input and output scaling and transformations. In section 5 the concept of the distance based learning method is introduced, and together with the methods of centroid shaking, biased classification, and principal components learning are described and the experimental results are reported. In section 6 the latest work on analysing learning system VC-dimension and network generalisation are discussed. Section 7 is the conclusion and discussion.

2. BiDirectional Neural Networks

A new method of training neural networks is suggested that enables them to remember input patterns as well as output vectors, given either of them. They may be considered to be associative memories, and are capable of classification, prediction, and finding cluster centroids.

Bidirectional associative memories (BAM) (Kosko, 1988, 1992) and the bidirectional version of Counter-propagation networks (Hecht-Nielsen, 1986) have been developed to take advantage of the possible advantages of bidirectional mappings. Remaining faithful to the simplicity and capability of the error Back-Propagation algorithm (BP), however, BDNNs (Nejad and Gedeon, 1994) avoid many of the serious problems these earlier models of hetero-associative bidirectional attractors or hybrid networks demonstrate such as capacity, efficiency, or multiple learning. Moreover, BDNNs have the power to determine expected cluster centroids, at a desired level of optimisation for each class, while minimising the effects of outliers.

To use BDNNs as content addressable memories, it has been assumed that a complete set of training data with a one-to-one relation between input and output vectors is available, otherwise some data preparation techniques may be used to create a one-to-one relationship between existing input-output data, unless the problem is inherently not appropriate for such transformations of the vector spaces. A one-to-one relationship is not necessary for finding cluster centres.

We apply the error back-propagation technique in both reverse and forward directions to adjust the weight matrix of the network. In our experiments we did not need to use more hidden units or more hidden layer weights in training a bidirectional network in comparison to the case of training a network in the traditional way. Therefore, preventing any increment in network size, we avoid any increment in VC-dimension of the network which would lead to a decrease in generalisation ability of the learning system.

As mentioned, the networks have been trained bidirectionally with the same weights in each direction. For each input and output node a threshold is assigned, but assigning extra threshold units to hidden layer units is not necessary. That is, the same threshold weights are used in both directions. Due to a flatter search space, usually a higher number of epochs will be necessary for the network to converge in comparison to those of traditional single direction back-propagation networks.

3. Data Sets

The following data sets have been used in our experiments. Due to space constraints, only the results on the first data set are quoted where these results are consistent with the other data sets.

3.1. Students Final Mark Prediction (SFMP) data set consists of 150 samples. Each pattern has fourteen input attributes. Four output have been used to classify the marks. Each record comprises student information, semester assessments without the final exam component and the subsequent final mark for a sample of students from a first year Computer Science subject at the University of New South Wales (Gedeon and Turner, 1993).

3.2. Geographical Information Series (GIS) data set consists of some satellite information from 190 samples from a rectangular grid of 24494 points. Each pattern has seventeen input attributes and one output. 143 records have been used in the training set and 47 records have been used as unseen data. The satellite information has been collected, augmented, and preprocessed in the School of Geography in University of NSW to classify a large geographical area in some forest supra-type categories (e.g. dry sclerophyll and wet sclerophyll) (Bustos and Gedeon, 1995).

3.3. Flight Simulation (FS) data set consists of 1800 records. Each records has 20 attributes. Outputs should predict the appropriate command in each time step to control automatic flight.

3.4. Gross Domestic Product (GDP) data set consists of 160 patterns. Each record has fourteen socio-economic indicator input attributes. They are used to predict the amount of GDP for a specified developing country (Gedeon and Good, 1993).

3.5. Artificial data sets with three inputs and one output to estimate the function $Y = \alpha X_1 + \beta X_2 + \gamma X_3$. Training samples of size 1331 have been deliberately generated to cover all the possible range of independent variables for 343 multivariate equations (Nejad and Gedeon, 1995).

4. Introduction to Data Representation

Each problem requires its own representation. Neural networks with more than one output should estimate an appropriate value for each output. Multi-Task Learning (MTL) will be done if there is more than one output. Input and output data are either real or binary depending on the problem and learning model. For unique concepts binary attributes are usually more appropriate. Representations are either local or distributed. In a local representation, only one attribute is on at a time. In a distributed representation a set of attributes could be on at the same time. A distributed representation has the advantage of using less attributes, thus smaller size network. Normalisation and scaling are common approaches to encode the raw data. The most common scaling technique used is (x - LowerBound) / Range. Mathematical transformations such as Fast Fourier Transform (FFT) and Wavelets are strong tools to capture the complexity of large size inputs. This is done frequently in tasks such as image processing.

5. Suggested Representation Methods

In this section we discuss a number of techniques for data representation. The first two methods use the cluster centres found using our BiDirectional method.

5.1 Distance Based Learning

5.1.1. Methodology

BDNNs were used to find the expected cluster centroids of the desired data set. These centroids are more reliable than the mean vector of input patterns for a specific class (Nejad and Gedeon, 1994). Outliers can significantly diverge the centroids. BDNNs are more reliable because they will adapt to the desired level of optimisation; thus, minimising the effects of the outliers.

Suppose c_k is the cluster centroid of class k, and p_i is the i^{th} input pattern. $p_{ik} = (p_i - c_k + 1)/2$ was used to measure and scale the distance of each record from cluster centroids of SFMP data set. Two extra units were used to determine the class of p_i and the related centroid of k. This increased four times

the number of input examples. We trained the network to learn these distances. In the recall phase, new patterns were produced for each test pattern using the same formula to measure distances from cluster centroids. According to the level of generalisation appropriate thresholds were used to decide the class of unseen patterns under uncertainty.

5.1.2. Experimental Results

We applied this method on SFMP data set. We randomly selected 70 records as training data. Fifty unseen records were selected to measure the performance. Applying this distance bases learning method reduced generalisation error of the test data set from 34% to 18%. That is, the performance increased from 66% to 82%.

5.2. Centroid Shaking Method

5.2.1. Methodology

Abu-Mostafa shows how we can take advantage of prior knowledge about the relationship between the input features and the function we want to learn. BDNNs can be used to extract some implied knowledge (e.g. invariance, monotonicity, and approximation hints) about the function we want to estimate. Once these hints are extracted using the positive and negative instances, it is possible to use these kinds of information to enhance learning system ability.

BDNN's cluster centroids were used to determine the most expected value of the attributes of input patterns for each class. Then some small amount of noise was added to each attribute. When pre-knowledge of the problem is not available other methods should be used to confirm that the changed attributes do not exceed the allowed ranges. Causal index, hidden index, or sensitivity analysis methods can be used as appropriate methods to assure this. These methods assign a significance factor to each input. The actual amount of noise should be inversely related to the significant of each attribute. Then these new records will be added to our training set to cover a wider variety of the problem space.

Some statistical methods may be used to produce the new exemplars. For example, we could calculate the standard deviation and mean value of the attributes. Then, *shaking* the records may be done by perturbing each variable with respect to its standard deviation.

5.2.2. Experimental Results

An experimental study on SFMP data set, by producing 20 extra *shaken* centroid exemplars, showed 10% increment in generalisation ability of the trained network, from 66% to 76%, which is not as good as the previous method. Figure 1 shows the input units significant indexes of SFMP data set obtained by three methods. These indexes have been used to determine the shaking bounds of each independent variable.

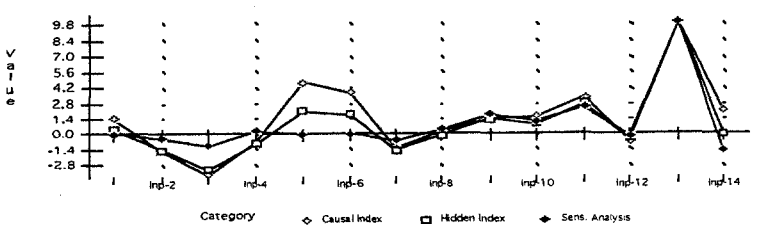

Figure 1. Comparing Hidden Indexes vs CI and SA approaches applied to a trained neural network with five hidden units to predict student final marks.

These methods are in general agreement, the simple perturbation based sensitivity analysis provides the least discrimination. The hidden index method is descibed elsewhere (Nejad and Gedeon, 1995).

5.3. Biased Classification Method

5.3.1. Methodology

Although most researchers have reported the dependency of learning system generalisation on sample size, relatively little attention has been paid to the appropriate distribution of positive and negative examples among output classes. Even if we had a large number of input patterns, but not distributed appropriately in the feature space to cover all the possible states of input patterns of separate classes, the training network will be biased toward the classes with higher density. This section provides a solution to optimise the classification in this situation.

Suppose $t_j^k \in \{0.1, 0.9\}$ is the desired output of the j^{th} dependent variable of the k^{th} class, and x_i is the i^{th} input vector. We argue that in order to have an unbiased learning system t_j^k should be substituted by $t_j^k * \frac{p(c_k \mid x_i)}{p(c_j \mid x_i)}$, for all k and i. $p(c_k \mid x_i)$ is the posterior probability of class c_k given a sample x_i. This Baysian projection of output vectors will reduce the classification bias toward the classes with higher density.

When the negative examples of a class belong to more than one class, computing baysian probability needs to determine prior and conditional probability of each class by using expert domain knowledge. To be faithful to the notion of simplicity of programming in neural computation we suggest computing $t_j^k * \frac{p(c_k)}{p(c_j)}$ instead of $t_j^k * \frac{p(c_k \mid x_i)}{p(c_j \mid x_i)}$ only for *off* output units. To avoid meaningless values in the target and the slow convergence of fermi function in upper and lower bounds, it is preferable to assign lower bound .1 and upper bound .5 to target values (except where $k = j$). Therefore, if the output vector is [.9, .1, .1, .1] and the prior probability of the classes are .50, .20, .20, and .10 respectively, the transformed output vector will be [.9, .25, .25, .5].

5.3.2. Experimental Results

We transformed the output vectors of SFMP by the proposed method and repeated the training of the network. Evaluating the generalisation ability of the new method showed 82% against the 66% of traditional representation. This result is as good as the distance base learning method. It worth noting that in the cases of classification of continuous categories the contingency restriction may enhance this method. For example in the GDP data set we can restrict farther units from having a higher value than closer units. for example, if the transformed output vector is [.5, .2, .2, .9], the restricted transformation will be [.2, .2, .2, .9].

5.4. Principal Components Learning

5.4.1. Methodology

Principal Component Analysis (PCA) is a statistical method to solve the problem of multi-colinearity among a set of variables by extracting a small number of uncorrelated variables (principal components). The new variables (factors) are a linear combination of the original variables. That is, for j^{th} factor of the k^{th} sample is estimated as $\hat{F}_{jk} = \sum_{i=1}^{p} W_{ji} X_{ik}$, where X_{ik} is the original value of the j^{th} variable for sample k and W_{ji} is the factor score coefficient for the j^{th} factor and i^{th} variable.

This is the same task which is down by the hidden layer of a multilayer perceptron. Unfortunately, for the statistical method of PCA as well as feed-forward neural networks, goodness-of-fit tests for the adequacy of these factors are directly related to the number of examples. Thus, deliberately selecting the

weighted combinations using pre-knowledge of the problem domain is required. Inputting expert knowledge into a system, by defining appropriate partitions on the input vector space, may reduce the input dimension. Other kinds of transformations are also known to reduce the input dimension. FFT and wavelets are among these,. which are frequently used in image processing tasks. These techniques are able to reduce the complexity of the problem by capturing some part of the complexity in the preprocessing phase. This will let the network capture better and more efficiently the remained complexity.

5.4.2. Experimental Results

To show the multi-colinearity effect on the trained neural networks, we did some experiments with our artificial data sets. We studied a few networks which were trained to estimate the output of some multivariate equations by correlated inputs. The experiments showed that not only was the convergence slowed down, but the causal index and sensitivity analysis were unable to determine the correct significant effect of each variable in all cases.

Linear combinations were used to reduce the input dimension of the SFMP data set to five. We dropped the first and the second variables. A weighted linear combination of the lab marks, which are highly correlated and with large variance, and two other simple linear combinations reduced our input dimension from 14 to 5. Then a neural network with 3 hidden units and the same outputs as before was used to classify the classes. The result was an increase in performance to 82% from 66%. This was a great reduction in generalisation error, size of network, and convergence speed. One of the major advantages of this method is the smaller size of the network leads to easier analysis and meaningful and understandable rules can be more readily extracted.

Note that this method again increased the performance from the base of 66% to 82%, like two of the previous methods.

6. Generalisation and Vapnik-Chervonenkis Dimension

To understand better why our representation methods are able to have better performance on the patterns which have not been seen during training, we will use the concept of Vapnik-Chervonenkis (VC) dimension. VC dimension and Valiant's model for Probably Approximately Correct (PAC) learning have been utilised to measure the amount of capacity or complexity of representational methods.

These models agree that when sample complexity and computational complexity, and therefore learning system VC dimension increases, more examples will be needed to be learned accurately as training data to enable the system to represent a wide variety of approximation functions (Valiant, 1984; Blumer, 1989).

It has been shown that for the N-input single-layer network, the VC dimension of the network is N+1 (Baum and Haussler, 1989). If $m \geq O\left(\frac{W}{\varepsilon} \log \frac{N}{\varepsilon}\right)$, $0 < \varepsilon \leq \frac{1}{8}$, random examples can be loaded on a feed-forward network using linear threshold function with N nodes and W weights, so that at least a fraction $1 - \frac{\varepsilon}{2}$ of the examples are correctly classified, then one has confidence approaching certainty that the network will correctly classify a fraction $1 - \varepsilon$ of the future test examples drawn from the same distribution.

The Vapnik upper bound for the worst-case generalisation error is less than or equal to ε, for $\varepsilon \leq O\left(\frac{d}{m} \ln \frac{m}{d}\right)$ (Vapnik, 1982). Ehrenfeucht (1988) showed that ε for some classes is $\varepsilon \geq O\left(\frac{d}{m}\right)$. Haussler (1990) showed that for binary classification $\varepsilon \leq \frac{d}{m}$. Recently, it was shown that in some cases, the average generalisation is significantly better than the VC bound (Cohn and Tesauro, 1992). There exist some tasks for which increasing network size improves generalisation (Amirikian and Nishimura, 1994). In fact, the VC-dimension of a feed-forward neural network with linear threshold gates is $\Omega(w.\log w)$ instead of $O(w.\log w)$ (Maass, 1994).

7. Discussion and Conclusion

The question of how complex learning a problem of interest by a given network remains an interesting problem still to be closely investigated. However, there is no doubt that an increase in W will usually result in an increase in VC-dimension, and generalisation performance significantly depends on the data representation method and problem specification.Therefore, in order to get better performance from a trained network we should either reduce W or increase the sample size.

Due to the nature of the problem and other restrictions, increasing the sample size is not always possible. Thus, it is suggested to reduce the network size and focus on data representation. There exist different methods of reducing the network size. A method of weight elimination is shown in Weigend (et al, 1991). Some researchers have suggested to reduce the number of hidden nodes and therefore the W (eg. Kruschke, 1988; Gedeon and Harris, 1991). Balancing bias and variance by determining the optimal (not necessarily the minimal) number of hidden units will result in the best performance of a specified network (Geman et al, 1992, Slade and Gedeon 1993).

We have used our BiDirectional neural networks model which we have shown elsewhere to be effective tools for classification, prediction, associative representation, and finding the centroid of a cluster. Here we have used the BDNNs to find the centres of clusters to address some parts of the data representation problem. We have used the centroids with our distance based learning approach, and with our centroid shaking approach to achieve good representations, which is a key factor in reducing the generalisation error. The biased classification and PCA methods require significantly more effort to use, and only achieve the same level of results. For example, for best results, the PCA approach may require expert domain knowledge. Nevertheless, the approach we have presented provided a different benefit – the smaller network can be more easily analysed in terms of rules.

In summary, we have used a variety of data sets from real-world problems to support the benefits of our methods, and evaluated the relationship of these methods to statistical learning theory. Our results are consistent with the Vapnik-Chervonenkis bounds, thus our methods can be considered as efficient means of designing the required bias in solving dynamic and complex learning systems and to increase their expected performance. This is by enhancing data representation and partitioning effectively the feature space and reducing generalisation error.

Acknowledgments

We would like to thank Claude Sammut from the A.I. Group in the School of Computer Science and Engineering at the University of NSW for giving access to the flight simulation data, and A.K. Skidmore from The Centre for Remote Sensing in the School of Geography at the University of NSW for giving access to the GIS data used.

References

Amirikian, B., and Nishimura, H. 1994. What Size Network Is Good for Generalisation of a Specific Task of Interest? Neural Networks, Vol. 7, No. 2, 321-329.

Anderson, J.A. 1988. Cognitive and Psychological Computation with Neural Models. *IEEE Transactions on Systems, Man, and Cybernetics 13*, 799-815.

Baum, E.B., and Haussler, D. 1989. What Size Net Gives Valid Generalisation? *Neural Computation 1*, 151-160.

Blumer, A., Ehrenfeucht, A., Haussler D., and Warmuth M. 1989. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM, 36(4)*, pp.929-965.

Bustos, RA and Gedeon, TD Decrypting Neural Network Data: A GIS Case Study, *Proceedings International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA)*, Alès, 1995.

Cohn, D. and Tesauro, G. 1992. How Tight are the Vapnik-Chervonenkis Bounds? *Neural Computation 4*, 249-269.

Ehrenfeucht, A., Haussler, D., Kearns, M., and Valiant, L. 1988. A General Lower Bound on the Number of Examples Needed for Learning. In *Proceedings of the 1988 Workshop on Computational Learning Theory*. San Mateo, CA, Morgan Kaufmann.

Gedeon, T.D. and Bowden, T.G. 1992. Heuristic pattern reduction. *International Joint Conference on Neural Networks*, Beijing, pp. 449-453.

Gedeon, TD and Good, RP Interactive modelling of a neural network model of GDP, *Proceedings International Conference on Modelling and Simulation*, pp. 355-360, Perth, 1993.

Gedeon, TD and Harris, D Network Reduction Techniques, *Proceedings International Conference on Neural Networks Methodologies and Applications*, AMSE, vol. 1, pp. 119-126, San Diego, 1991.

Gedeon, T.D. and Turner, H. 1993. Explaining student grades predicted by a neural network. *Proceedings International Joint Conference on Neural Networks*, pp. 609-612, Nagoya.

Geman, S., Bienenstock, E., and Doursat R. 1992. Neural Networks and the Bias/Variance Dilemma. *Neural Computation 4*, 1-58.

Haussler, D., Littlestone, N., and Warmuth, M. 1990. Predicting {0,1}-Functions on Randomly Drawn Points. *Technical Report UCSC-CRL-90-54*, University if California at Santa Cruz.

Hecht-Nielsen, R. 1987. Counterpropagation Networks. *Applied Optics*, vol. 26, no. 3, 4979-4984.

Kosko, B. 1988. Bidirectional Associative Memories. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-18, 49-60.

Kosko, B. 1992. Neural Networks and Fuzzy Systems.

Kruschke, J.K. 1988. Creating Local and Distributed Bottlenecks in Hidden Layers of Back-Propagation Networks. *Processing of the 1988 Connectionist Summer School*. Carnegie-Mellon University, Pittsburgh, PA: Morgan Kaufmann.

Maass, W. 1994. Neural Nets with Superlinear VC-Dimension. *Neural Computation 6*, 877-884.

Nejad, A.F., Gedeon T.D. 1994. BiDirectional MLP Neural Networks. *International Symposium on Artificial Neural Networks*. Tainan, Taiwan, pp. 308-313.

Nejad, A.F., Gedeon T.D. 1995. Analyser Neural Networks. *International Workshop on Applications of Artificial Neural Networks to Telecommunications*. Stockholm, Sweden.

Rumelhart, D.E., Hinton, G.E., Williams, R.J. 1986. Learning internal representations by error propagation. in Rumelhart, D.E., McClelland, *Parallel Distributed Processing*, Vol. 1, MIT Press.

Slade, P. and Gedeon, T.D. 1993. Bimodal Distribution Removal, *Proceedings IWANN International Conference on Neural Networks*, Barcelona. also in Mira, J., Cabestany, J. and Prieto, A. 1993. *New Trends in Neural Computation*, pp. 249-254, Springer Verlag, Lecture Notes in Computer Science, vol. 686.

Valiant, L.G. 1984. A Theory of the Learnable. *Communications of the ACM*, 27, 1134-1142

Vapnik, V.N. 1982. Estimation of Dependencies Based on Empirical Data. Spring-Verlag, New York.

Weigend, A., Rumelhart, D., and Huberman, B. 1991. Generalisation by Weight Elimination with Application to Forecasting. In *Advances in Neural Information Processing Systems 3*, R. Lippmann, J. Moody, and D. Touretzky, eds. Morgan Kaufmann, Denver, Co.