An Information Theoretic Methodology for Prestructuring Neural Networks

Bjorn Chambless, George G. Lendaris, Martin Zwick
NW Computational Intelligence Laboratory, Portland State University
P.O. Box 751, Portland, OR 97207, USA
bjorn@cs.pdx.edu, lendaris@sysc.pdx.edu, zwick@sysc.pdx.edu

Abstract

Absence of a priori knowledge about a problem domain typically forces use of overly complex neural network structures. An information-theoretic method based on calculating information transmission is applied to training data to obtain a priori knowledge that is useful for prestructuring (reducing complexity) of neural networks. The method is applied to a continuous system, and it is shown that such prestructuring reduces training time, and enhances generalization capability.

1 Introduction

In the absence of significant *a priori* knowledge about the problem to which an artificial neural network (ANN) is to be applied, an appropriate starting structure for the ANN would typically include full interconnection between adjacent layers. Once the ANN paradigm and number of layers is selected, the choice of full interconnection provides the maximum capability of representing/modeling the target system. Unfortunately, this most general configuration also maximizes the dimensionality of the weight space, giving rise to the dual problems of increased training time and likely convergence to an inferior local-minimum in weight space (in the sense of providing poorer generalization for the resulting ANN).

For typical multivariate systems, the relation to be modeled is not fully general; that is, the relation is likely to be decomposable into simpler sub-relations, since not all system variables will exhibit the same degree of interdependence.

In this paper, an information-theoretic approach is used to perform an analysis of the data from the problem context in order to determine the structure of the relation to be modeled. Then, the knowledge obtained concerning the *mathematical* structure of the problem is translated into a form that allows an appropriate *physical* prestructuring of the ANN. The objective of the prestructuring is to reduce the dimensionality of the weight space in a principaled way, while reducing training time and improving generalization capability.

This is not the first work to use information-theoretic methods in conjunction with neural networks (e.g. see [3]). A closely allied method to the current work has been applied to discrete systems [9][8][7] however this may be the first application of information-theoretic neural prestructuring to continuous systems.

2 Bias/Variance Justification for Pre-structuring

The modeling process may be abstracted to approximating the observed relation $X \rightarrow Y$ (X and Y may be vectors or scalars) with a function $f(X)$ for the purpose of predicting Y for future observations of X. The ideal approximation will produce the most likely Y for a given X, so the optimal $f(X)$ is

$$\widehat{f}(X) = E_Y[Y|X]$$

Error resulting from the conversion of a potentially *stochastic relation* into a *function* may be measured using the squared difference between the observed output and the optimal function approximation $\widehat{f}(X)$.

$$E[(Y - \widehat{f}(X))^2] \tag{1}$$

In practice, a model will be constructed from a finite sample of data observations, D. A specific model, developed using a data set D, from the set of feasible models will be designated $f_D(X)$.

Error for a specific model f_D is then

$$\varepsilon = E[(Y - f_D(X))^2] = E[(Y - \widehat{f}(X))^2] + (f_D(X) - \widehat{f}(X))^2 \tag{2}$$

The first term on the right side of Equation 2 is a measure of the non-deterministic component of the relation $X \rightarrow Y$ (Equation 1), or the stochastic variance of Y given X, and is not reducible.

Therefore, the performance of a modeling methodology can only be improved by reducing the contribution of the second term. To accomplish this, we use

$$E_D[(f_D(X) - \widehat{f}(X))^2]$$

For the NW Computational Intelligence Laboratory and IJCNN 2001, Washington D.C.

where E_D indicates the *expected value over all possible data sets D*. This error may be further decomposed as

$$E_D[(f_D(X) - \widehat{f}(X))^2] = (E_D[f_D(X)] - \widehat{f}(X))^2 + E_D[(f_D(X) - E_D[f_D(X)])^2] \quad (3)$$

This equation separates error due to the difference between the possible models and the optimal function $\widehat{f}(X)$, into *bias* error and *variance* error terms. The first term on the right of Equation 3 is the square of the difference (over all data sets D) between the expected value of the output of the models generated using specific data sets, D, and the expected output Y given X (the ideal deterministic model $\widehat{f}(X)$). This is the expected value of the *bias* of the model. A large bias error term (in relation to the variance term) suggests the model chosen is poorly suited to the data sets, D. The model is too simple (more parameters are required) and/or the model is fundamentally incompatible with the system (e.g. a linear model applied to a non-linear problem).

The second term is the degree of variability of the models over the possible data sets. I.e., the expected value (over all the data sets) of the squared difference between the average model and the model generated by a given data set. This is the tendency of the model to vary across the possible data sets. It is the component of the error due to *over-fitting*, or in the ANN context, *over-training*. A large variance error value (relative to bias error) typically indicates that the model has too many parameters (weights for ANNs) for the data sets. Or, equivalently, the data sets are too small and/or noisy for the complexity of the model.

The relationship between these error components is such that efforts to decrease one tend to increase the other, giving rise to the *bias/variance dilemma* [2]. For example, adding parameters (complexity) to a model will likely reduce the bias error since the increased capability of the model will allow it to better approximate $\widehat{f}(X)$, but this increased power will also allow the model to better fit any noise present in D.

A solution to the dilemma is to purposefully introduce non-destructive bias into the model which allows the variance, $E_D[(f_D(X) - E_D[f_D(X)])^2]$, to be reduced without increasing the bias error, $(E_D[f_D(X)] - E_y[Y|X])^2$. With selected knowledge of $\widehat{f}(X)$, prestructuring may be accomplished by fixing parameters in the model in a manner which facilitates the approximation of $\widehat{f}(X)$. By fixing these values, the number of free parameters is reduced, thereby helping to reduce the variance error (the tendency to over-fit the data). Neural networks may be so prestructured, by either fixing or removing (equivalent to setting to 0) selected connection weights.

Speaking of the importance of this approach, Geman, et al.[2] suggest that, " ...learning complex tasks is essentially impossible without the a priori introduction of carefully designed biases into the machine's architecture."

The modeling process must therefore incorporate some degree of meta-level knowledge about $E_Y[Y|X]$ (the ideal function approximation, $\widehat{f}(X)$). However this raises the question: How can such knowledge be discovered? and the apparent paradox: Since "black-box" modeling methods such as ANNs are most useful in contexts where little of the internal dynamics of the target system are known, the knowledge of the system required to make the application of neural modeling techniques successful would appear to be the same knowledge which ensures that these techniques are not needed. A method is described and demonstrated in the following sections that (successfully) addresses these issues.

3 Information-Theoretic Analysis

The proposed method provides a *non-parametric* means of identifying *relevant* sub-relations within a larger target system. The analysis may be performed using the same data[1] which will subsequently be used to train the neural network to model the target system.

3.1 Computation of Information-Theoretic Measures

From the *law of statistically independent events*, the probability of two events p and q occurring in the same trial is $p \cdot q$ *if and only if* p and q are statistically independent. Without assuming the distributions of these events, we can suppose that the *degree of dependence* between events can be measured as the *divergence* of the joint distribution of these events from the independence distribution. This divergence, for example between nominal variables X and Y, can be quantified using *information transmission*, $T(X:Y)$[2].

$$T(X:Y) = H(X) + H(Y) - H(XY)$$

where $H(X)$ and $H(Y)$ are the Shannon entropies [10] of the marginal distributions of X and Y.

$$H(X) = -\sum_i \left(\sum_j p_{ij}\right) \log_2 \left(\sum_j p_{ij}\right)$$

$$H(Y) = -\sum_j \left(\sum_i p_{ij}\right) \log_2 \left(\sum_i p_{ij}\right)$$

where p_{ij} is the probability of state i, j occurring in the joint distribution of X and Y, where i designates the state of X and j is the state of Y.

[1] The data must be clustered prior to analysis. See Section 4.2.

[2] Transmission is also known as *mutual information*.

$H(XY)$ is the entropy of the joint distribution,

$$H(XY) = -\sum_{i,j} p_{ij} \log_2 p_{ij}$$

By measuring transmission between variables, dependencies within the system can be identified, and more importantly, if the dependence between variables is negligible, relations may be partitioned into simpler subrelations

Information-theoretic entropy is the measure of *uncertainty* of a distribution ($H(X)$ is maximum when X is uniform and minimized when only one state for X is observed). Transmission $T(X:Y)$ is then the degree of constraint in the joint distribution not accounted for by the combined uncertainties of X and Y.

These methods for detecting constraint differ from common statistical measures such as the *correlation coefficient* and *covariance* which assume either the distribution of the data or the nature of the constraint (e.g. linear). Since measurements of information transmission are non-parametric, transmission is sensitive to *any* constraint between observed variables. This property makes the technique appealing as an analysis tool applied prior to a "black box" modeling methodology such as neural networks for which it is difficult to predict which facets of a system the network will model successfully.

3.1.1 Statistical Significance for Information Theoretic Models. Information theoretic transmission values are tested for statistical significance using the standard χ^2 test. Degrees of freedom for the test are computed as the difference in degrees of freedom between the proposed model and the degrees of freedom for the joint distribution (See Equation 4). For the transmissions computed in the example, these values are listed as "df" and "χ^2 sig." in Table 1. Degrees of freedom for a transmission is calculated as follows

$$df_{T(XY)} = df_{XY} - df_X - df_Y \quad (4)$$

A limiting factor in this type of analysis is statistical significance. Since χ^2 significance declines with the growth in degrees of freedom, higher order distributions where the constituent variables have a large number of states may not yield significant results.

4 An Example System

Given a directed system of 5 variables: A,B,C,D, and E, where A,B and C are inputs for independent variables (IVs) and D and E are outputs for dependent variables (DVs), the lattice of general structures[3] (Figure 1) shows the possible models of the system. A general structure may be transformed into a specific structure by assigning the variables: A, B and C to the inputs and the variables: D and E to the outputs.

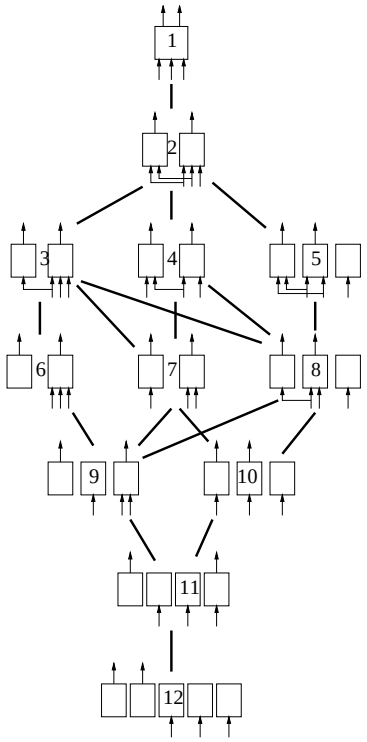

Figure 1: Lattice of Structures for a 5-variable directed system

The lattice is a hierarchy with the *saturated* (non-decomposable) model at the top (for this example system, this is structure 1 in Figure 1). The descendants are simplified models, constructed by partitioning relations. The extreme decomposition exhibits complete independence between the variables. For this example system, this is structure 12 in Figure 1. Each descendant in the lattice is the result of a single partition between an IV and a DV in the parent. For example, structure 2 is derived from structure 1 by breaking the dependence of one of the DVs on one of the IVs. Since one output now only depends on two of the three inputs, the single 3-input, 2-output system is transformed into two subrelations where one output is only dependent on two of the inputs.

4.1 Definition of the System Relations

The example system is defined to be a relation composed of two sub-relations. These sub-relations are the functions [4]

[3] These are the distinct structural forms. In contrast, a specific structure may be shown to be isomorphic to another specific structure by the reassignment of variables. It should be noted that this terminology is not universal.

[4] Since the system is deterministic, the error term given in Equation 2 will be due solely to error from Equation 3. No noise was added to the data.

$$D = \sin(A+B)$$
$$E = \sin(B+C)$$

Since A does not affect E, and C does not affect D, this system corresponds most closely to structure 4 (in Figure 1). A,B and C are restricted to be real values in the interval $[0, \pi]$. To produce observations of this system, 1000 random input triples were generated.[5] These inputs were coupled with the corresponding system outputs to produce the 1000 system observations for analysis and network training.

4.1.1 Discussion of the Choice for Sub-relation Functions. The *sine* function of summed inputs was chosen in order to demonstrate the capacity of the method to cope with a high degree of non-linearity. Furthermore, the same function was used over the same domain and range to show the capacity of the analysis method to distinguish between variables from similar distributions. The complexity of the system is just within the capacity of the network size chosen, so any destructive bias introduced should render the network incapable of modeling the target system successfully.

4.2 Analysis of Observations

Since the information-theoretic methods employed require discrete distributions, continuous variable observations must first be converted to nominal values. For this system, a simple uniform clustering algorithm[6] was applied. However, for a more complex data set, a less naive approach would likely improve the results of the analysis. If the variables were nominal, no clustering would be required[7] Results of the information-theoretic analysis, using a clustering granularity of 5 clusters per dimension, are given in Table 1.

Transmission values do depend on clustering granularity, but the system dynamics are clear from the results of even this coarse clustering. As expected, for the dependent variable D, $T(C:D)$ is trivial while $T(A:D)$ and $T(B:D)$ are sizable. $T(AB:D)$[8] accounts for 67% of the uncertainty of the dependent variable. The transmission value of $T(ABC:D)$ is larger, but the increase is small (0.0792 bits) and since $T(C:D)$ is negligible, it is apparent that this small increase is due to sampling error.

For output E, $T(B:E)$ and $T(C:E)$ are both large and

[5] A,B and C inputs come from a uniform distribution over the interval $[0, \pi]$.

[6] The algorithm "rasterizes" the continuous data by dividing the domain and range into N equal intervals.

[7] Unless increased statistical significance is required.

[8] This should be read as the information transmission between the joint distribution AB and the dependant variable D.

	Trans.	$\%H(DV)$	df	χ^2 sig.
$T(A:D)$	0.4013	18.292	16	1.0
$T(B:D)$	0.3992	18.194	16	1.0
$T(C:D)$	0.0082	0.373	16	.213
$T(AB:D)$	1.4691	66.960	96	1.0
$T(AC:D)$	0.4433	20.205	96	1.0
$T(BC:D)$	0.4426	20.176	96	1.0
$T(ABC:D)$	1.5483	70.573	496	1.0
$T(A:E)$	0.0125	0.566	16	0.636
$T(B:E)$	0.3921	17.753	16	1.0
$T(C:E)$	.3633	16.450	16	1.0
$T(AB:E)$	0.4405	19.946	96	1.0
$T(AC:E)$	0.4102	18.570	96	1.0
$T(BC:E)$	1.4795	66.983	96	1.0
$T(ABC:E)$	1.5691	71.041	496	1.0

Table 1: Information transmissions using 5 clusters

$T(BC:E)$ has the largest significant transmission. And, as expected, the relations between C and D (expressed as $C:D$) and between A and E(expressed as $A:E$) may be disregarded.

4.2.1 Traversal of the Lattice of Structures. It is clear from the discounted relations: $A:E$ and $C:D$, that structure 4 from Figure 1 is the most accurate model for the example system. Figure 2 shows the path of traversal down the lattice corresponding to the two relations eliminated. For each IV to DV relation discounted, one step is taken down the lattice. The order that relations are removed from the model is not relevant since the resulting structure will be the same. Based on the above analysis, no other relations may be be disregarded, so no further simplification is possible.

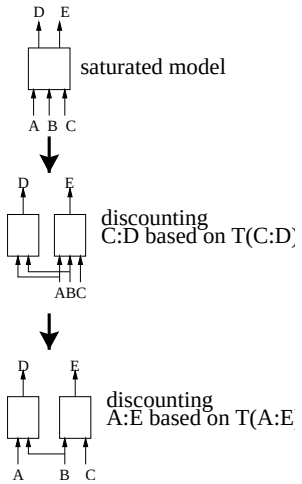

Figure 2: Traversal of the lattice for the example system. The supporting transmission analysis is show in Table 1

It should be noted that Reconstructability Analysis

(RA) provides a much more systematic way of traversing the lattice of structures and determining "goodness of fit" for a given model. Unfortunately, a complete description of the RA methodology is beyond the scope of this paper. The reader is directed to [11], [6] and [5].

5 Translating the Model into a Prestructured Neural Network

A single hidden layer neural network with 3 inputs, 2 outputs and 8 hidden layer nodes was deemed adequate to model the example system.[9] The fully connected configuration contains 40 connection weights (not including biases). Selected weights were pruned from the fully connect configuration, using the selected model shown in Figure 2 as a "template." This results in the network shown in Figure 3. Since the relations $A:E$ and $C:D$ were determined to be negligible, any connections between the input A and the output E, and the input C and the output D can provide only "crosstalk".

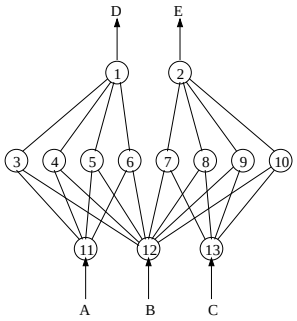

Figure 3: Prestructured 3-8-2 feed-forward network

Since the internal "complexity"[10] of each of the subrelations is unknown, equal numbers of hidden layer nodes are allocated for each. The resulting "biased" configuration is such that input A is connected to 4 hidden layer nodes, and input C is connected to the other 4. B is fully connected to the hidden layer since B participates in both identified sub-relations. From the hidden layer, the 4 nodes connected to input A are connected to the output node D and the remaining 4 notes (which are connected to input C) are connected to output E. The biased network contains 24 connection weights, a 35% reduction in connection parameters.

6 Comparison Between the Biased and Unbiased Networks

Typical responses for D and E against the ideal are shown in Figures 4 and 5 for an *unbiased* network and in Figures 6 and 7 for a *biased* network. In Figure 4, the horizontal axis is $x = A + B$, so the ideal response

[9] The activation functions are sigmoid and the training algorithm is the *delta-bar-delta learning rule* (EDBD) [4].

[10] Complexity is used loosely in this context as an estimate of the ANN resources (hidden nodes) required to model the relation.

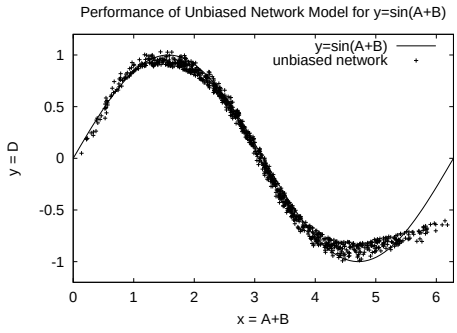

Figure 4: Comparison between fully-connected ("unbiased") net's response and ideal response for output D

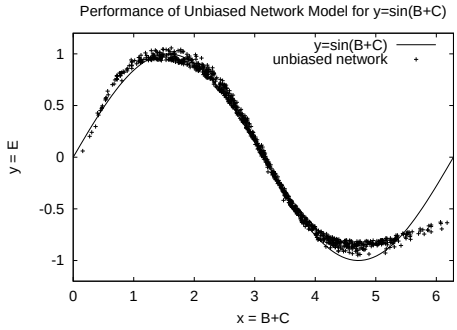

Figure 5: Comparison between fully-connected ("unbiased") net's response and ideal response for output E

$(D = \sin(A + B))$ is plotted as $y = \sin(x)$. There are numerous combinations of A and B that yield a specific value of x, and correspondingly, a specific value for $y = \sin(x)$. If the neural network function approximator were perfect, it would yield a clean sine wave output (at the D node) for all combinations of A and B at the inputs. Since the neural network involves multiple paths that combine in the output node, there is considerable opportunity for mapping error to occur, in particular, when (non-zero) connections from the C input node are allowed to impinge on the D neural element. The latter contribute a kind of cross talk component to the output value. Data for Figure 4 was generated by applying a large set of A, B, and C combinations to the input nodes of the neural network, and recording the value out of the B node. To plot the data, the A and B values were added to yield an x coordinate, and the corresponding output of the D node plotted on the y axis. This resulted in the scattered values of y on either side of the ideal sine wave plot. The key thing to observe is that the scattering is significantly lower in Figure 6 (prestructured/biased neural network) than in Figure 4 (fully-connected/unbiased neural network). Similar comments apply to dependent variable E in Fig-

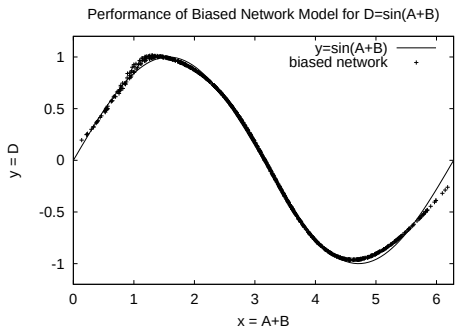

Figure 6: Comparison between prestructured ("biased") net's response and ideal response for output D

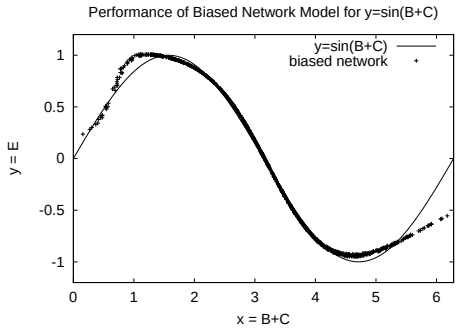

Figure 7: Comparison between prestructured ("biased") net's response and ideal response for output E

ures 7 and 5. Whereas in principle, the neural network could have learned to eliminate the cross-talk terms in its architecture, by virtue of manually deleting these terms (i.e. introducing purposeful bias) based on the *a priori* knowledge obtained via the information-theoretic analysis, the neural networks job was made substantially easier and less prone to error.

6.1 Comparison of Convergence

Quality of converge was compared by sampling RMS error after 500,000 training cycles. The network configuration used was the same as in section 6.2. Data from only 20 runs with each configuration was sufficient to demonstrate superior convergence for the prestructured network. The mean difference in RMS error was 0.005125. Applying the *Student's t* test of statistical significance, the difference in RMS error is found to be outside the 93.1% confidence interval for the differences of the means; suggesting a significant difference in quality of convergence. That is, there is a 93.1% probability that the two samples of RMS error are from different distributions, indicating that the convergence characteristics of the prestructured network are measurably better.

6.2 Comparison of Training Times

Training time was compared by sampling the number of training cycles to the point where RMS error for an epoch fell to 0.05 or below. The weight parameters were initialized to small random values and the networks were trained using the Extended Delta-Bar-Delta[4] algorithm. We conducted 41 sample runs for each network configuration using an epoch size of 100. The mean number of training cycles ($\overline{X}_{cycles}$) to achieve $RMS \leq 0.05$ was 13,662 for the unbiased configuration and 11,072 for the prestructured configuration. Applying the *Student's t* test to the difference in the means as was done in Section 6.1, the probability that the difference is statistically significant is overwhelming (>99.999%).

7 Conclusion

In previous work ([7][8][9]), Reconstructability Analysis[5][6][11] has been applied to the context of discrete systems to yield *a priori* information useful in prestructuring neural networks with the objective of reducing train time, and enhancing generalization capability. The material of the present paper extends the approach to the context of continuous systems.

References

[1] Ashby, R., "Constraint Analysis of Many-Dimensional Relations" , *Prog. in Bio-Cyber. II*, Wiener and Schade, 1965

[2] Geman, S., E. Bienenstock, and R. Doursat, "Neural Networks and the Bias/Variance Dilemma", *Neural Computation*, MIT Press, Vol. 4, pp 1-58, 1992

[3] Hertz, J., K. Anders, R. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, 1991

[4] Jacobs, R.A., "Increased Rates of Convergence Through Learning Rate Adaptation", *Neural Networks*, Vol. 1, pp295-307, 1988

[5] Klir, G. R. Cavallo, "Reconstructability Analysis", *Int. Journal of General Systems*, Vol. 17, pp33-61, 1981

[6] Krippendorff, K., *Information Theory: structural models for qualitative data*, Sage Pub., Newbury Park, CA, 1986

[7] Lendaris, G., K. Mathia, M. Zwick, "On Matching ANN Structure to Problem Domain Structure", *Proc. of the World Congress on Neural Networks*, 1993

[8] Lendaris, G., K. Mathia, "Using A Priori Knowledge to Pre-structure ANNs", *Australian Journal of Intelligent Information Systems*, vol 1, no 1, March 1994

[9] Lendaris, G., T. Shannon and M. Zwick, "Prestructuring Neural Networks via Extended Dependency Analysis with Application to Pattern Classification", for SPIE'99, Orlando FL, Jan. 1999

[10] Shannon, C.E., "A Mathematical Theory of Communication", *Bell Sys. Tech. Journal*, No. 27, 1948

[11] Zwick, M., "Wholes and Parts in General Systems Methodology", *Evolutionary Biology and Characteristics*, Academic Press, 1999