COMMITTEE NETWORKS BY RESAMPLING

JANET M. TWOMEY
Department of Industrial and Manufacturing Engineering
Wichita State University
Wichita, KS 67260

ALICE E. SMITH
Department of Industrial Engineering
University of Pittsburgh
Pittsburgh, PA 15261

ABSTRACT:

Artificial neural networks (ANN) are nonparametric models and like all nonparametric models require a large number of observations to build and evaluate the model. But data is always finite and most often scarce in real world applications. The question now becomes: How does a modeler build an ANN model on a limited sample size and obtain an accurate and reliable estimate the model's error? The research presented in this paper proposes a novel method for building and evaluating ANN under the condition of scarce data: Committee networks by resampling.

INTRODUCTION

There are two commonly used methods of artificial neural network (ANN) model building and evaluation: One method is to subdivide the total available sample of n observations into a training set of size n_1 and a test set of size n_2; $n = n_1 + n_2$. A single network is constructed of n_1 observations and is tested for generalization capabilities on n_2 observations. This method is the **train-and-test** (T/T) method. The expected error of the ANN model on a new observation is estimated by (1), where $\hat{f}(T_{n1}, x_j)$ is the network constructed on n_1 training pairs, L is a loss function (mean absolute error or mean squared error), and T_{n1} is the training set used to construct the ANN model.

$$\hat{E}rr_{T/T} = \frac{1}{n_2}\sum_{j=1}^{n_2} L[y_j, \hat{f}(T_{n1}, \mathbf{x}_j)] \quad (1)$$

Because ANN are nonparametric models [1] they require large amounts of data to achieve the best performing model. Therefor in cases where samples sizes are small, the ANN model built on a reduced number of observations, n_1, will not perform as well as an ANN built on n observations. The second method builds and evaluates the ANN model on all n observations. While utilizing all n for model building, this method tends to underestimate the error of the ANN model because the same data is used to for both training and testing. This method known as the **resubstitution** method, is given by (2), where $\hat{f}(T, x_j)$ is the network constructed on n.

$$\hat{\mathrm{E}}\mathrm{rr}_{\mathrm{Resub}} = \frac{1}{n}\sum_{i=1}^{n} L[y_i, \hat{f}(T, x_i)] \tag{2}$$

When the data available to the modeler is unlimited, both methods are adequate. However, in most real world problems data is not unlimited, therefore building and evaluating an ANN is more problematic. The research presented in this paper addresses the question of - How to build an ANN model on all available data and obtain an accurate and reliable estimate of the model's expected error under conditions of scarce data? A novel approach is presented that combines the committee network (CN) method of ANN model building with the cross-validation (CV) and bootstrap methods of ANN model evaluation. This paper begins with a brief introduction to committee networks and network evaluation through resampling. Experimental results from 100 monte carlo simulations of the new approach are presented, conclusions are drawn and future work is discussed.

Committee Network (CN) Approach to ANN Model Building

The committee network approach to ANN building has also been referred to as "majority fusers" or "democratic networks" [2,3]. It is compared to the more commonly used single network approach. There are numerous variations on the committee network theme, but the general idea is as follows: A number of networks, k, are built and trained to solve the same problem, where the problem is either classification or function approximation. An input vector $\mathbf{x_j}$, is provided to each trained network:

$$\textit{trained net}_i(\mathbf{x}_j) \rightarrow y_{ij}; \text{ for } i = 1, 2, \ldots, k. \tag{3}$$

where i is the ith network, and j is the jth input vector. The outputs y_{ij} of each network are combined using some rule or fuser, *f*. The output, Y_j, that is the result of the rule is then used as the final prediction:

$$f(y_{ij}) \rightarrow Y_j; \; i = 1, 2, \ldots, k. \tag{4}$$

The performance of the committee network is evaluated by comparing Y_j to target t_j, following the usual train-and-test methodology The theory behind committee networks is that a composite system of combined agents will be better than the best of the agents that went into the composite system. One premise of committee network training is that the k networks can be trained in any fashion without regard to over or under fitting of the data, and that the error of the k networks is unimportant. Methods to combine the system of agents for classification purposes have been proposed [2,3,4]. Hashem and Schmeiser [4] have investigated an optimal linear combination fuser of trained neural networks to solve function approximation problems.

Resampling Methods for ANN Model Evaluation

Three resampling approaches for estimating the error of a prediction model were considered in this research: cross-validation, group cross-validation and bootstrap methods. They are the preferred methods for estimating the error of a statistical prediction rule, but their use in ANN has been limited [1,5,6]. **Cross-validation** (CV) uses data resampling to obtain an estimate of a model's error that has been constructed on all n observations [7]. More specifically, the estimate of the model's error is determined by constructing n models on sub-sets of all data points leaving out one (size n-1), then testing on the one omitted point (size 1) in all n possible ways (5), where j is the excluded point and $T_{(j)}$ is the data set minus point j used to construct the model $\hat{f}(T_{(j)}, \mathbf{x}_j)$.

$$\hat{E}rr_{CV} = \frac{1}{n}\sum_{j=1}^{n} L[y_j, \hat{f}(T_{(j)}, \mathbf{x}_j)] \tag{5}$$

A variation of CV is group cross-validation (GCV), where H points are excluded G times. Thus, n = GH. G models are built to estimate Err.

The **bootstrap method** was originated by Efron [8]. Like cross-validation it resamples n observations to estimate the error of an ANN model built on all n observations. Bootstrap data sets are created via resampling with replacement, where cross-validation data sets are created by resampling without replacement. To obtain $\hat{E}rr_{Boot}$, let $\overline{e}rr$ be the training set error of the of ANN model built on all n training pairs. The bootstrap method estimates the expected excess error, $\hat{\omega}_{Boot}$ of $\overline{e}rr$. $\hat{\omega}_{Boot}$ is assessed through independent bootstrap training sets T^{*1}, T^{*2}, ..., T^{*B}; where B is number of bootstrap samples. For each T^{*b}, a prediction model is constructed, $\hat{f}(T^{*b}, \mathbf{x}_j)$. The excess error for a bootstrap sample b is given by (7):

$$\omega_b^* = \frac{1}{n}\sum_{i=1}^{n} L\left[y_i\,,\ \hat{f}\left(T^{*b}, \mathbf{x}_i\right)\right]^2 - \frac{1}{n}\sum_{i=1}^{n} L\left[y_i^*\,,\ \hat{f}\left(T^{*b}, \mathbf{x}_i^*\right)\right]^2 \tag{7}$$

$\hat{\omega}_{Boot}$ is approximated by averaging over all B:

$$\hat{\omega}_{Boot} = \frac{1}{B}\sum_{b=1}^{B} \omega_{Boot}^{*b} \tag{8}$$

The expected error of an ANN model according to the bootstrap method is then estimated by (9):

$$\hat{E}rr_{Boot} = \overline{e}rr + \hat{\omega}_{Boot} \tag{9}$$

COMBINED APPROACH OF COMMITTEE NETWORKS AND RESAMPLING METHODS

For this paper the ANN model that is ultimately applied to problem is referred to as the *application network*, while networks trained for error estimation purposes are referred to as resampled networks. The results presented in this paper are an exploration into *an ANN training methodology that enables the formulation of the best prediction model while also providing an accurate and reliable measure of the network's performance under conditions of scarce data.* Accordingly, two hypothesis were formulated:

1. Committee networks derived from the resampling validation networks will perform at least as well as any single application network built on the total available sample (resubstitution, cross-validation, group cross-validation, and bootstrap methodologies) and better than application networks built on reduced sample sizes (train-and-test methodology).

2. The resampling $\hat{\text{Err}}$ will estimate the Err of a committee network at least as well as it estimates the Err of single network, and better than $\hat{\text{Err}}$ that are based on reduced training and testing sets (train-and-test methodology).

Methodology and Experimental Design

A noisy function approximation problem was used for experimentation. Training pairs were generated according to (11), where $\varepsilon \to N(0,.2)$ and x ranges from 0.0 to 3.10:

$$y = 4.26(e^{-x} - 4e^{-2x} + 3e^{-3x}) + \varepsilon \qquad (11)$$

To compare methodologies network architecture, training algorithm and training parameters were held constant. Network size: 1 input node, 1 hidden layer of 3 nodes, and 1 output node. Network training algorithm: original backpropagation algorithm [9,10] with the addition of a momentum term. All network weights were randomly initialized to the same starting point. All training parameters were held constant with a sigmoidal transfer function, a training rate of 0.10, and momentum term of 0.90. Networks continued to train until the network no longer continued learn as determined by the change in weights. Results and conclusions are based upon one hundred research trials. A single trial proceeds as follows. One random sample, of size n, T_n , is generated according to (11). T_n is the *total sample available* to the modeler for building and evaluating the ANN. From T_n, ANN models are built and tested according to the five error estimation methodologies (train-and-test, resubstitution, cross-validation, group cross-validation, and bootstrap). Four methods (resubstitution, cross-validation, group cross-validation, bootstrap), build the application network on T_n. Thus for a single trial, the same application network is used for those four methods. Conversely, the T/T method constructs the application network on a randomly chosen sub-sample of n_1 training pairs. Therefore, a separate application network was built on T_{n1} for the T/T method. An approximation to the *true error*, Err, of each application network was obtained from 5000 independent observations sampled from the same population. Estimates of Err, were determined according to the five methodologies.

According to (3) and (4), committee application networks were constructed by fusing together the resampled networks that were trained to obtain $\hat{\text{Err}}_{CV}$, $\hat{\text{Err}}_{GCV}$, and $\hat{\text{Err}}_{Boot}$. The Err of the committee application networks were determined from the same 5000 independent observations. Three committee network fusers, *f*, were examined: mean, 10% trimmed mean and median. The best fuser was found to be the mean since it produced the CN with the lowest Err.

Three levels of n where investigated, n = 5, 10 and 20. To control for the effects of random sampling of size n, the single trial described above was replicated 100 times using different randomly sampled T_n, at each level of n.

Experimental Results

With regard to Hypothesis 1: Overall, committee application networks did not significantly out-perform application networks comprised of a single network trained on T_n. However, the committee network approach did out perform the single network approach trained on reduced samples sizes (train-and-test method). The 95% C.I. for the Err for n = 5 shown in Figure 1 reveal that committee application networks tend to perform better and with less variability than networks built using the single network approach. Figure 1 also reveals that committee networks built using the bootstrap method performed better than committee networks built using the cross-validation methods. This trend continues at n = 10. As the total available sample size increases to n=20, application networks using either the CN or the single network approach built on training set size > 18, perform $\pm$ 0.05 of the expected Err, of each other.

With regard to Hypothesis 2: The mean squared error (MSE) of the error estimate, $\hat{\text{Err}}$, is used to measure the degree to which an individual $\hat{\text{Err}}$ differs from Err of an application network built using the single network approach or of a application network using the CN approach. The MSE of the $\hat{\text{Err}}$ are provided in Figure 2. The Figure indicates improvement in the bootstrap $\hat{\text{Err}}$ when it is used to estimate the Err of a committee network built using bootstrap resampling; i.e. the MSE is smaller within a narrower error bar. As n increases to 20, all estimates perform equally well within $\pm$ 0.02 of each other.

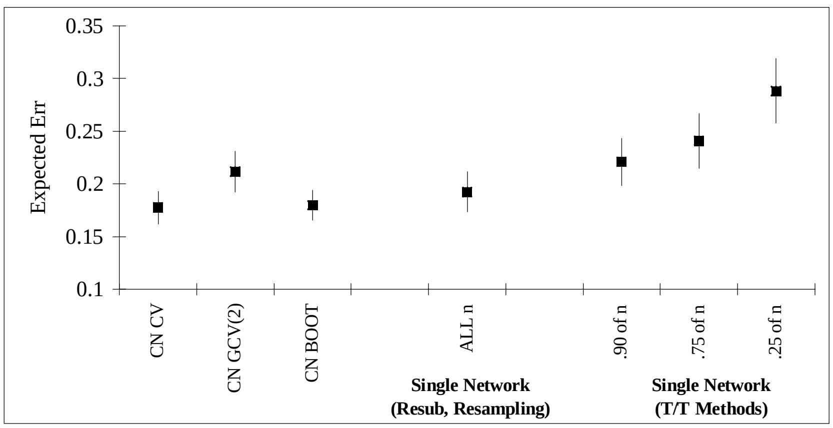

Figure 1: 95% C.I. of Err of Application Networks: n=5

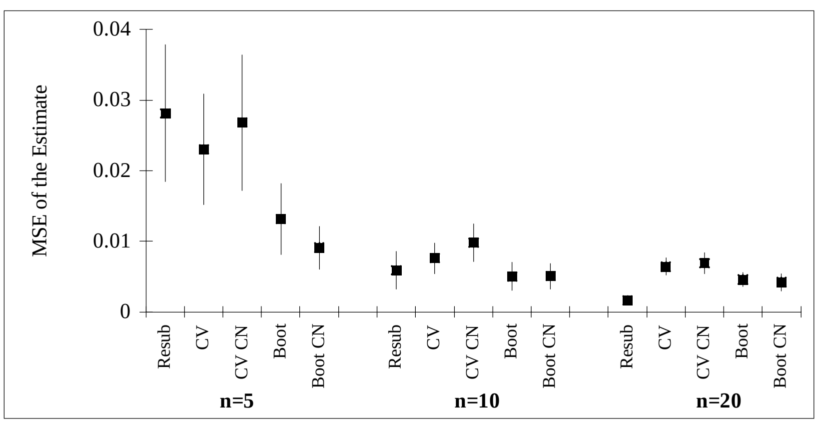

Figure 2: 95% C.I. of the MSE of $\hat{\text{Err}}$

Discussion Of Results And Future Research

The results indicate application networks constructed under the committee network approach performed as well as application networks built under the single network approach, and had a reduction in variability of Err. The results also indicate the resampling $\hat{\mathrm{Err}}$ can be a better estimate of the Err of a CN, than of the Err of a single network built on all n observations. The problem examined here, noisy approximation problem of a single input and a single output is relatively simplistic. The benefits of this approach are more likely seen in more complex approximation cases like the real world problem of this research. This approach does appear to be a promising method for utilizing total available sample for training purposes. Other topics for future research in the area of combining the committee network approach with the resampling methods include:

1. Committee networks constructed according to the bootstrap method performed better than the cross validation methods. Is this due to the number of times the data is resampled (number of networks) or the resampling plan (with or without replacement?
2. Other methods of constructing and fusing the committee network, but still utilizing the resampling approaches for validation and error estimation should be explored; e.g. optimal combination (5).

REFERENCES

1. Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, **4**, 1-58.
2. Battiti, R. and Colla, A.M. (1994). Democracy in neural nets: voting schemes for classification. *Neural Networks*, **7**, 691-707.
3. Rogova, G. (1994). Combining the results of several neural network classifiers. *Neural Networks,* **7**, 777-781.
4. Hasham, S. and Schmeiser, B. (1992) Improving model accuracy using optimal linear combination of trained neural networks. *Technical Report SMS92-16*, School of Industrial Engineering, Purdue University.
5. Twomey, J.M., Smith, A.E., and Redfern, M.S. (1995). A predictive model for slip resistance using artificial neural networks. *IIE Transactions,* **23**, pp. 374-381.
6. Twomey, J. And Smith, A. E. Nonparametric error estimation methods for the evaluation and validation of artificial neural networks. *Intelligent Engineering Systems Through Artificial Networks: Volume 3, ASME Press, 233-238.*
7. Lachenbruch, P. and Mickey, M. (1968). Estimation of error rates in discriminant analysis. *Technometrics*, **10,** 1-11.
8. Efron, B. (1982). The jackknife, the bootstrap, and other resampling plans. *SIAM NSF-CBMS, Monograph*, **38.**
9. Werbos, P.J. (1974) *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences,* Ph.D. in Statistics, Harvard University.
10. 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, Vol. 1: Foundations.* D. E. Rumelhart and J.L. McClelland, and the PDP group, Eds., MIT Press, Cambridge, MA, 318-362.