

# Nonparametric Error Estimation Methods for Evaluating and Validating Artificial Neural Network Prediction Models

Janet M. Twomey and Alice E. Smith

*Department of Industrial Engineering*

*University of Pittsburgh*

*Pittsburgh, PA 15261*

*aesmith@engrng.pitt.edu*

## **ABSTRACT:**

Typically the true error of ANN prediction model is estimated by testing the trained network on new data not used in model construction. Four well-studied statistical error estimation methods: cross-validation, group cross-validation, jackknife and bootstrap are reviewed and are presented as competing error estimation methodologies that could be used to evaluate and validate ANN prediction models. All four methods utilize the *entire* sample for the construction of the prediction model and estimate the true error via a resampling methodology.

## **INTRODUCTION**

The evaluation and validation of an artificial neural network (ANN) prediction model is based upon some error function; usually mean squared error, mean absolute error, or if appropriate, percent incorrect classification. Since the objective of a prediction model is to predict successfully on new data, the *true error* of a model is statistically defined on "an asymptotically large number of new data points that converge in the limit to the actual population distribution" [1]. In most real world applications unlimited sample sizes are impossible or too costly to obtain. As a consequence the network modeler is faced with the "performance/evaluation" dilemma; on one hand, much data is needed to build or train the network model but on the other hand much data is also needed to get an accurate evaluation of the model. Most modelers opt for achieving a better performing model at the expense of a good evaluation. The focus of our current research seeks to answer the question; Can the true error of an ANN prediction model be empirically extrapolated from limited sample sizes?

Typically the true error of an ANN prediction model is estimated by testing the trained network on new data not used in model construction. In cases where data is severely limited this procedure is not always performed. Consequently, the true error is estimated using the same data that was used to construct the model. This paper reviews four well-studied statistical error estimation methods: cross-validation, group cross-validation, jackknife and bootstrap. All four methods utilize the *entire* sample for the construction of the prediction model and estimate the true error via a resampling methodology. These methods are currently used to evaluate and validate statistical prediction models. They have been shown to provide good error estimates, but can be computationally very expensive in terms of the number of prediction models constructed. The purpose of this paper is to present them as competing error estimation methodologies that could be used to

evaluate and validate ANN prediction models. Some experimental results are given.

### THE PREDICTION PROBLEM

Consider a class of prediction models,  $f$  (statistical or ANN), so that for any given  $\mathbf{x}_k$  we can predict  $y_k$ . Assume that there is some unknown distribution  $F$ , from which a random sample,  $T$  (training set), is drawn:  $T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$ ; where  $t_i = (\mathbf{x}_i, y_i)$  and  $t_i \stackrel{iid}{\sim} F$ .  $T$ , of sample size  $n$ , is selected from a population of size  $N$ ; where  $n \ll N$ . Construct a model,  $\hat{f}(T, \mathbf{x})$ .  $\hat{f}(T, \mathbf{x})$  is then used to predict  $y_k$  from  $\mathbf{x}_k$ , where  $t_k = (\mathbf{x}_k, y_k)$  (test set) is not used in the construction of the prediction rule  $\hat{f}(T, \mathbf{x})$ . How good is  $\hat{f}(T, \mathbf{x}_k)$  (for  $k \in [1, N]$ )?

The measure of the prediction error is determined according to some specified loss function  $L$ . The squared error will be used throughout this paper. The true error,  $Err$ , is the expected value over the entire population:

$$Err = \frac{1}{N} \sum_{i=1}^N [y_i - \hat{f}(T, x_i)]^2 \quad (1)$$

The true error could be calculated if we knew  $y_k$  and  $\hat{y}_k$  for all  $k \in [1, N]$ . Since this is unlikely, it is necessary to quantify the error of the model by obtaining an estimate of the true error,  $\hat{Err}$ . One obvious estimate of true error is the error of the sample,  $T$ , used to construct the model. According to Efron [2, 3] this error is known as apparent error :

$$\bar{err} = \frac{1}{n} \sum_{i=1}^n [y_i - \hat{f}(T, x_i)]^2 \quad (2)$$

Note that in general  $\bar{err}$  is biased downward ( $Err \geq \bar{err}$ ), since the same data was used to both construct and test the model. Therefore the true error equals the apparent error plus bias ( $\beta$ ):

$$Err = \bar{err} + \beta \quad (3)$$

and thus:

$$\beta_{TRUE} = Err - \bar{err}. \quad (4)$$

To correct for this bias, the standard (train-and-test) ANN method or one of four resampling methods can be used to estimate the excess error. Excess error is a good indicator of model over-parameterization.

### EXCESS ERROR ESTIMATION METHODOLOGIES

This section reviews five methods for estimating the true excess error of a prediction model. While the origin and the vast majority of research into the various error estimation methodologies have come from the area of traditional statistics (regression, logistic regression, etc.), these methods have occasionally been 'naively' applied to the area of ANNs. Due to the iterative and stochastic nature of ANN model construction there is no reason to assume that the results from the statistical literature can be directly applied to ANNs.

**1. Apparent error methodology.** In cases where the ANN model is constructed on all available data ( $n$ ), the true error of the model is estimated by:

$$\hat{E}_{err} = \bar{e}_{err} = \frac{1}{n} \sum_{i=1}^n [y_i - \hat{f}[T, x_i]]^2; \quad (5)$$

where bias is estimated to be:  $\hat{\beta} = 0$ . Since only one network is constructed, this method is the computationally one of the least expensive.

**2. Test-and-train methodology.** The standard ANN procedure is to take a finite number of samples (n) and randomly divided them into two sub-samples of data; sub-sample -1, (n<sub>1</sub>): for model construction (training),  $\hat{f}(T_{n1}, \mathbf{x})$ ; and sub-sample-2, (n<sub>2</sub>): after the model has been constructed, for model evaluation and validation (testing). Test sets have typically included anywhere from 5% to 90% of the total number of available points. In cases where data is severely limited, the amount of data proportioned to training/testing becomes a more critical question. According to train-and-test the bias is estimated by:

$$\hat{\beta}_{T-T} = \frac{1}{n_2} \sum_{j=n_1+1}^n [y_j - \hat{f}[T_{n1}, x_j]]^2 - \frac{1}{n_1} \sum_{i=1}^{n_1} [y_i - \hat{f}[T_{n1}, x_i]]^2. \quad (6)$$

This method is computationally very inexpensive since only one network is built.

**3. Cross-validation and group cross-validation.** There are a few researchers and practitioners that use the group cross-validation [4, 5, 6] approach to ANN model evaluation and validation. In general this method removes a sub-sample of data (of size k) from the entire data set (size n); it then trains the network on the remaining n-k data points and tests on the k points left out. The sub-sample of data is then added back into the training set. Another sub-sample (size k), which does not contain data from the first sub-sample, is removed. A second network is trained on the remaining n-k points. This procedure is repeated until all n points have been removed and n/k networks have been constructed. If k=1 this method is simply known as cross-validation. Some practitioners select the final network from the n/k networks that were constructed. However, the correct procedure is to construct the final network on all n samples and estimate the true error via the cross-validation networks. For k=1, where j is the excluded point and T<sub>(j)</sub> is the data set minus point j, the training set T<sub>(j)</sub> = {(x<sub>1</sub>, y<sub>1</sub>), (x<sub>2</sub>, y<sub>2</sub>), ..., (x<sub>j-1</sub>, y<sub>j-1</sub>), (x<sub>j+1</sub>, y<sub>j+1</sub>), ..., (x<sub>n</sub>, y<sub>n</sub>)}, is used to construct the model  $\hat{f}[T_{(j)}, x_j]$ . The bias

according to cross-validation is estimated by [7]:

$$\hat{\beta}_{cv} = \frac{1}{n} \sum_{j=1}^n [y_j - \hat{f}[T_{(j)}, x_j]]^2 - \frac{1}{n} \sum_{i=1}^n [y_i - \hat{f}[T, x_i]]^2. \quad (7)$$

This methodology is computationally more expensive since n/k + 1 networks must be constructed. However it has potential for providing a better model since all n points are used to construct the final model.

**4. Jackknife methodology.** The jackknife estimate of bias is very similar to the group cross-validation method. The only difference is in the way that it assesses apparent error; apparent error is averaged over all n/k prediction models. Bias is estimated by [8]:

$$\hat{\beta}_{JACK} = \frac{1}{n} \sum_{j=1}^n [y_j - \hat{f}[T_{(j)}, x_j]]^2 - \frac{1}{n} \sum_{i=1}^n \frac{1}{n} \sum_{j=1}^n [y_i - \hat{f}[T_{(j)}, x_i]]^2. \quad (8)$$

The final network is built from all n sample points; n/k + 1 models are constructed.

**5. Bootstrap methodology.** Although it has been shown to provide very good estimates of error for statistical prediction models, there are very few instances in the ANN literature where the bootstrap method of error estimation has been

applied to ANN prediction models<sup>[8]</sup>. This is most likely due to the increase in computational effort of building additional models. According to the originator of the bootstrap, Efron, it is the maximum likelihood estimate of the true error [2,3]. The bootstrap method constructs the final model on all n data points and estimates the bias via resampling. Bootstrap samples are constructed as follows. Let  $\hat{F}$  be the empirical distribution function for T; where  $\hat{F}$  puts mass of 1/n on  $t_1, t_2, \dots, t_n$ . Let  $T^{*b}$  (bootstrap sample) be a random sample of size n taken *with replacement* from  $\hat{F}$ . The  $\hat{\beta}_{BOOT}$  is assessed through independent bootstrap training sets  $T^{*1}, T^{*2}, \dots, T^{*B}$ , where B is the total number of bootstrap samples. For each  $T^{*b}$ , a prediction model is constructed,  $\hat{f}_{T^{*b}, X_i}$ . The bias is estimated by[8]:

$$\beta_b^* = \frac{1}{n} \sum_{i=1}^n \hat{f}_{T^{*b}, X_i} - \hat{f}_{T, X_i} \quad (9)$$

and

$$\hat{\beta}_{BOOT} = \frac{1}{B} \sum_{b=1}^B \beta_b^* \quad (10)$$

where as  $B \rightarrow \infty$ ,  $\hat{\beta} \rightarrow \text{true bias}$ . Efron recommends (for statistical models) that B be not less than 25 and for practical purposes not greater than 200. B + 1 models are constructed; B models to estimate true bias, plus one final model.

### ASSESSMENT AND APPLICABILITY OF RESAMPLING METHODS

Efron compared the cross-validation, jackknife and bootstrap methods for estimating the excess error of a statistical prediction model. Efron [2,3] demonstrated, all three resampling approaches provide 'good' estimates of true bias and in turn give improved estimates of true error over the apparent error of a statistical prediction model. According to Efron:

1. Cross-validation gives a nearly unbiased estimate of Err but with very high variability (especially when n is small).
  2.  $\hat{\beta}_{JACK}$  gives very close results to  $\hat{\beta}_{CV}$ , which is not surprising since they are very similar.
  3. The bootstrap gives a downward biased estimate of Err but is less variable.
- There are benefits and costs of using these methods for both ANN and statistical prediction models. These methods are nonparametric or data driven. The benefits of nonparametric methods in general are : a) they demand minimal amounts of modeling; b) they require few assumptions or analysis; c) they are mechanistic or easy to apply for universal application; d) computing power is substituted for theoretical analysis. Whether or not  $\hat{\beta}_{CV}, \hat{\beta}_{GCV}, \hat{\beta}_{JACK},$  or  $\hat{\beta}_{BOOT}$  are better than  $\hat{\beta}_{T-T}$  at estimating the true error for ANNs is an open question. Theoretically, they should be at least as good as  $\hat{\beta}_{T-T}$ . Moreover, all of these error methodologies should give better estimates of true error over the apparent error. The greatest benefit associated with the resampling methodologies is their utilization of all n samples in model construction. The costs associated with the resampling approach to error estimation are obvious: computational costs. All three methods can require numerous recomputations of the prediction rule. With the advent of high speed computers, the recomputation of the statistical prediction rule sustains minimal costs. This, however, is *not* true for ANN models. Depending upon the network

architecture, training algorithm and training set size, a single network may require an inordinate amount of computing time to converge to a pre-specified tolerance. The trade-off between the costs of computational effort and the benefits of improved error estimates with an optimal utilization of the sample, are the subject of current research.

## EXPERIMENTAL RESULTS

The results of a small example problem are presented here in order to explore some of the issue raised above. One-thousand and ten (x, y) observations,  $N=1010$ , were generated according to:  $y = 4.26(e^{-x} - 4e^{-2x} + 3e^{-3x})$ ; x ranged from 0.0 to 3.10 (See Fig.1). Ten observations were randomly selected to make up the training and testing sample,  $n=10$ . The remaining 1000 observations were set aside as the *true* estimate of excess error. The prediction models, feedforward multilayer networks, were all constructed of 1 input node (x variable), 1 hidden layer of 3 nodes and 1 output node (y); all fully connected. All networks were trained using the standard backpropagation algorithm [9,10]. Training was terminated when either all training pairs were deemed correct within 10% of the targeted output, or after 10,000 presentations of the training sample. The loss-function for testing was based on mean squared error function (MSE). The standard ANN method of train-and-test was simulated first. Twenty percent (2 observations) of the sample was removed for testing and the remaining 80% (8 observations) was used for training. To examine the variability of the bias estimate, 45 networks, each with a different combination of train/test (8/2) samples, were trained and tested according to the method described above. The estimate of bias was calculated according to Eq. 6. True bias was evaluated for each of the 45 networks over the 1000 observations. The results, Table I, indicate that model bias estimated according to the train-and-test methodology,  $\hat{\beta}_{T,T}$ , on average over estimated the true model bias and is highly variable. Considering the 10 observations chosen as the total sample for training/testing, this result is not surprising. It does illustrate the potential problems associated with the standard ANN methodology; that is - model performance can be highly dependent on which observations are chosen for training/testing. The cross-validation, jackknife and bootstrap methods were also examined using the same sample of 10 observations. For a cross-validation of  $k=1$ , bias was estimated using Eq. 7. For a jackknife of  $k=1$ , bias was estimated using Eq. 8. In order to make the computational effort comparable, 0 bootstrap samples ( $B=10$ ) were constructed and the bias estimated according to Eqs 9 and 10. Thirty additional bootstrap samples ( $B=40$ ) were constructed to examine the effects of increased B. True bias was evaluated over the 1000 observations. The results are shown in Table II. The apparent error estimate of bias under estimated true bias, however it was the closest estimate and only required the construction of single network. The bootstrap method ( $B=40$ ) gave the next closest estimate of bias, but with the highest computational effort of 41 total networks constructed. Additional work is needed to assess the variability, and the behavior of the various estimates under a variety of conditions; e.g. sample size.

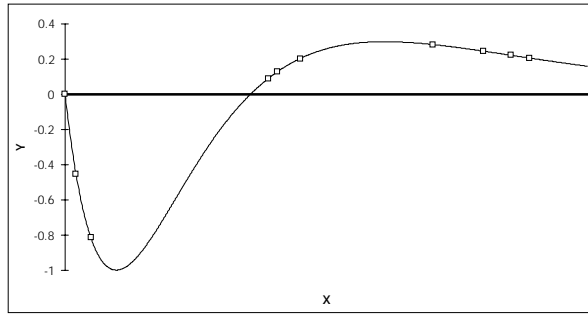


Figure 1. Plot of generated data; boxes indicate sample observations.

	<b>apparent error</b>	$\hat{\beta}_{T-T}$	$\beta_{TRUE}$	<b>abs(bias of <math>\hat{\beta}_{T-T}</math>)</b>
<b>Mean</b>	0.016	0.075	0.028	0.070
<b>Variance</b>	4.3E-05	0.016	0.001	

Table I. Results of Train-and-Test methodology.

	<b>True</b>	<b>App.</b>	<b>CV (k=1)</b>	<b>Jack (k=1)</b>	<b>Boot (B=10)</b>	<b>Boot (B=40)</b>
<b>Bias</b>	0.013	0.00	0.065	0.058	0.037	0.034
<b># Networks</b>		1	11	11	11	41

Table II. Estimates of bias.

## REFERENCES

1. Weiss, S., M., and Kulikowski, C.A. (1991). Computer Systems that Learn. Morgan Kaufmann Publishers, Inc., San Mateo, CA.
2. Efron, B. (1983). Estimating the error rate of a prediction rule: Improvement over cross-validation. *Journal of the American Statistical Association*, **78**, 316-331.
3. Efron, B. (1982). The jackknife, the bootstrap, and other resampling plans. *SIAM NSF-CBMS, Monograph*, **38**.
4. Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, **4**, 1-58.
5. White, H. (1990). Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings. *Neural Networks*, **3**, 535-549.
6. Twomey, J.M., Smith, A.E., and Redfern, M.S. (in press). A predictive model for slip resistance using artificial neural networks. *IIE Transactions*.
7. Gong, G. (1986). Cross-validation, the jackknife, and the bootstrap: Excess error estimation in forward logistic regression. *Journal of the American Statistical Association*, **81**, 108-113.
8. Welch, R.M., Sengupta, S.K., Goroch, A.K., Rabindra, P., Rangaraj, N., and Navar, M.S. (1992). Polar cloud and surface classification using AVHRR imagery: An intercomparison of methods. *Journal of Applied Meteorology*. **31**, 405-420.
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.