The Use of Cross-Validation in Neural Network Extrapolation of Forest Tree Growth

C. Gordon
Applied Mathematics
University of the Witwatersrand
E-mail: CHRIS@gauss.cam.wits.ac.za

Abstract

A back-propagation multilayer Artificial Neural Network (ANN) is used to model the mean growth rate of different plots of Pine trees. Cross-validation is found to be essential in providing a good extrapolation. Predictions are made for different plots which have different initial densities and different artificial thinning regimes. Comparisons are made with a standard nonlinear regression solution.

1 Introduction

Artificial Neural Networks (ANNs) present a very flexible, robust and general method of modelling data. An ANN can be trained on experimental data to model a nonlinear mapping realized by some system. One of the advantages of an ANN model over traditional nonlinear regression approaches is that it is not necessary to first find a suitable parametric form of the regression model for the problem at hand. A one hidden layer multilayer perceptron can act as a universal function approximator, [3].

It is necessary is to specify how many hidden layer neurons to use. The principle of Occam's Razor, [5], is a good guide to follow. A trade off between having the minimum number of hidden layer neurons and a good enough approximation to the training data needs to be made. The ratio of the number of weights to adjust in the ANN relative to the amount of training data should be kept as small as possible in order to provide good generalization performance to unseen data.

Training of the ANN should not necessarily try to provide the best fit possible. Training data often consists of noise plus signal. Also, it is not desirable to model the training data too closely as it will have features specific to it which do not provide information on the model as a whole. Fitting the training data too closely can lead to overfitting the data and seriously degrade generalization performance, [6].

Cross-validation provides a useful way of avoiding overfitting. Here part of the training set is set aside to monitor the generalization ability of the ANN. Training is stopped when the predicted generalization ability reaches a maximum, see Section 6.

In the rest of this paper these considerations will be illustrated on the problem of modelling forest tree growth. This problem will involve generalization to data not within the convex hull of the domain of the training data. That is the ANN will have to extrapolate the forest tree growth to future values. It will be seen that in this case avoiding overfitting is essential in order to provide good generalization ability.

2 Statement of Problem

The growth of a tree can be effected by competition from neighbouring trees for the available resources of sunlight, moisture, root space and soil nutrients, [9]. The old South African Forestry Division initiated an experiment in the 1930's to monitor the growth in supervised plots of Pine Trees (Pinus roxburghii) under different initial conditions and different treatments.

Twenty different plots were monitored. There were two controllable factors in each plot:

1. The initial number of trees in the plot. The trees were all planted at the same time and the intertree spacing is constant throughout the plot. An equivalent measure is the tree density which is equal to the number of trees per unit area.
2. The number of trees artificially removed from the plot each year. This process is known as *artificial thinning* and is performed so as to reduce the intertree competition.

Each plot was subjected to a different variant of these two conditions. Measurements were only taken on a subset of each plot. A representative section of about eight hundred square meters was monitored. Every few years the trees in this area had their diameters at breast height (DBH) measured. The experiment lasted from 1936 to 1973.

In this article we will only be concerned with the average DBH of the plot at a given time. Figures 1 shows an example of the mean growth data for a particular plot.

Along the x–axis is the age of the plot of trees (all the trees were planted at the same time) and on the y-axis is the average DBH taking over all the trees in the plot. The + are the data points. Each data point is contained in a bar whose height is proportional to the number of trees in the plot when the data point was recorded. The actual number of trees is displayed above the corresponding bar. The growth curve can be affected by a reduction in the number of trees. The reasons for this are that the mean is taking over a different number of trees and the trees have less competition after a reduction in their number. This external and discontinuous influence complicates the modelling process.

It is useful for the forestry manager to have some means of predicting the future values of tree growth. Thus we seek to develop a way of using the available data for each plot to predict what the future values of the growth parameter will be. Traditionally, nonlinear regression is used for such problems. Thus, for comparison purposes it is important that we examine this approach.

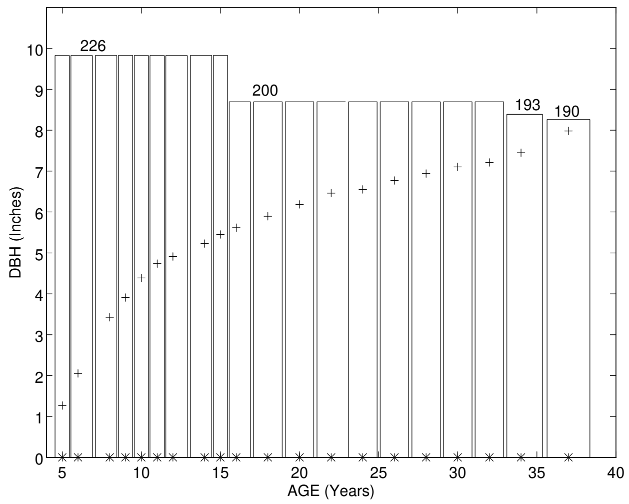

Figure 1: Plot1 DBH Data

3 Nonlinear Regression

Nonlinear models can be used for predicting and interpolating growth data. Both theoretical and empirical equations are used. Empirical equations are useful for interpolation, but can be unreliable for extrapolating. In order for an equation to be biologically plausible it must have an upper asymptote to express the fact that growth is limited.

Research has indicated that the best of the traditional nonlinear regression functions for modelling this data is the Schnutte function, [2]. The form of this function is:

$$Y(t) = \left[y_1^b + \left(y_2^b - y_1^b \right) \frac{1 - e^{-a(t-\tau_1)}}{1 - e^{-a(\tau_2-\tau_1)}} \right]^{1/b}$$

This function does have some biological motivation. Y is the target variable and it will be used to model the average DBH. The age is denoted by t. The arbitrary parameters are y_1, y_2, a and b. The conditions $a \neq 0$ and $b \neq 0$ must be satisfied. The variables τ_1 and τ_2 are constants chosen before fitting is begun.

The arbitrary parameters are found by minimizing

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (Y_i - d_i)^2 \,,$$

where d_i is the measured DBH at time t_i and i ranges over the training data.

The functional form of the Schnutte function is not ideal for numerical optimization, [7]. If good starting values are not chosen for the parameters then the Hessian matrix of the problem can be either close to singular or singular, thus making optimization difficult. However good starting values can be found as follows: Set τ_1 equal to the earliest training age and τ_2 equal to the latest training age. Have as the initial

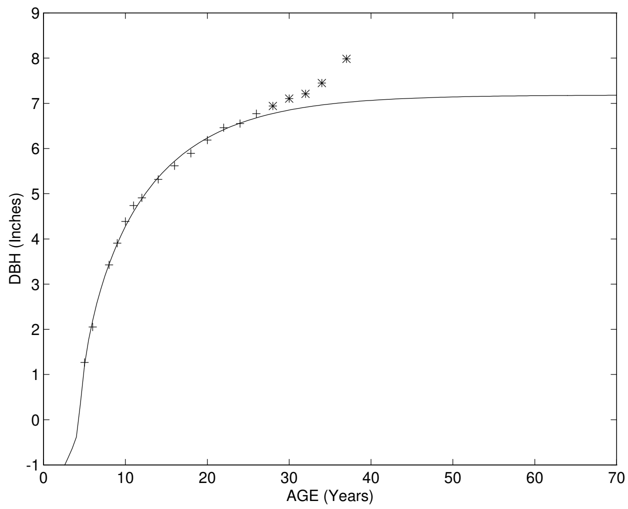

Figure 2: The Schnutte Prediction for Plot1. The + are the data points used to determine the parameters of the Schnutte Equation. The $*$ are the data points to be predicted.

guesses of y_1 and y_2, the corresponding measured DBHs at τ_1 and τ_2. Set the initial guesses of a and b at some nonzero value, eg 1.

The results of applying this scheme to the first 10 points of data shown in Figure 1 is shown in Figure 2. The + denote training points and the $*$ denote the test points, ie the data which was not used in any way to determine the optimum parameters for the Schnutte function. As can be seen, a good fit is obtained on the training data, but the function does not predict the correct future growth rate. The generalization ability of the Schnutte function appears poor in this case.

4 ANN Tree Growth Prediction

We now examine how an ANN can be used to predict forest tree growth. First, the architecture of the ANN has to be specified. The input layer needs only one input neuron for the age of the tree, t. The output neuron needs only one output neuron for the average DBH of the tree, h. From the Universal Approximation Theorem, [4], we know that it is sufficient to have only one hidden layer of neurons in order to be able to approximate any continuous, nonlinear mapping. As for the number of neurons needed in the hidden layer, it is desirable to have the least number of neurons (and therefor parameters) possible so as to avoid the problem of overfitting. But, enough neurons are needed so that the ANN is capable of providing a good enough approximation to the various nonlinear mappings for the different growth curves of the different plots. One hidden layer neuron was found too restrictive in the range of nonlinear mappings it could reasonably approximate, while two hidden layer neurons were found more than adequate for this purpose.

Figure 3 shows a diagram of the architecture of the ANN which was used. Note that the squares in this figure represent input nodes that serve the purpose of merely

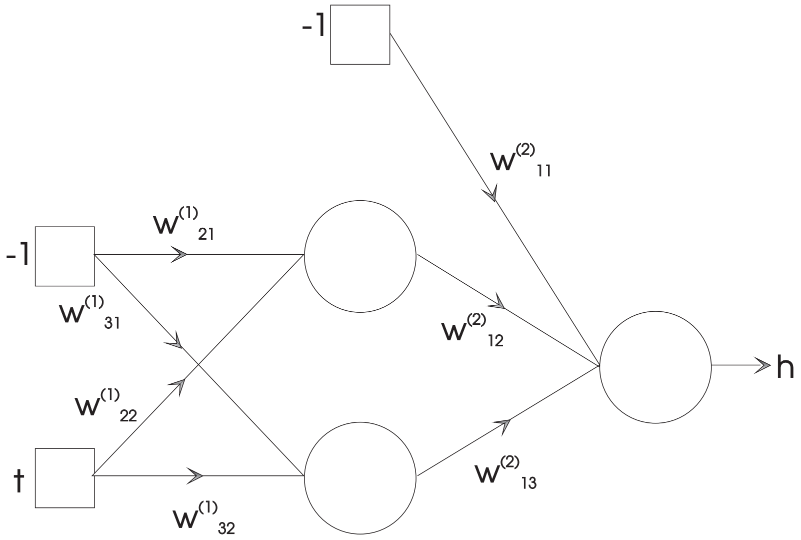

Figure 3: Three Layer ANN used for forest tree prediction.

inputting the data to the proceeding layer's neurons unmodified. While the circles are neurons that output a weighted sum of their incoming data, acted on by a transfer function.

As for the transfer function, there are basically two choices: A sigmoid function or a linear function, ie no transfer function. It has been recommended, [3], that a linear activation function should be used in the output layer because a sigmoid function is asymptotically bounded above and so it requires that the possible outputs of the mapping have to be within this upper bound. However, it was found that a linear output activation function is much less stable than a sigmoid output activation function, and requires the learning rate parameter to be decreased as learning proceeds, if convergence is to be reached. Thus all activation functions were made sigmoid logistic functions.

The equation for the ANN used can easily be written out in full for this small case:

$$h = \varphi\left(w_{12}^{(2)}\varphi\left(w_{22}^{(1)}t - w_{21}^{(1)}\right) + w_{13}^{(2)}\varphi\left(w_{32}^{(1)}t - w_{31}^{(1)}\right) - w_{11}^{(2)}\right). \tag{1}$$

Note that the problem can now be stated as find the weights, w_{ij}, in the above equation such that the MSE on the training data is low and good prediction for future h values are obtained. The Back Propagation algorithm will be used to find the MSE, since it is known to be efficient at avoiding local minima, [3].

There is still the problem of data scaling to be considered. The input data cannot be too large in magnitude or else the hidden layer neurons might be saturated. The change in weights during training is proportional to the gradient of the activation function. For large magnitude input values, the gradient of the sigmoid function is small and thus weight modification is slow. To avoid this the input data is scaled such that its maximum and minimum values (including the maximum value used for prediction) lies between -3 and 3. Since the output is restricted to lie between 0 and 1 for the logistic sigmoid activation function, the data output variables have to be scaled within this range.

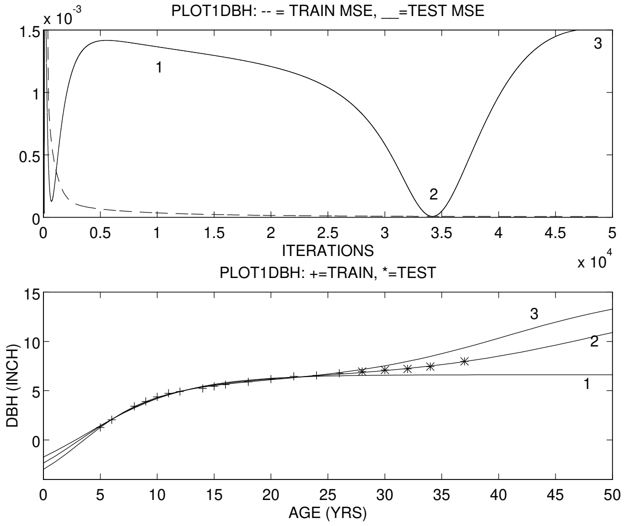

Figure 4: Top: The MSE of the training data (dotted line) and the MSE of the test data (solid line) as the iterations increase. Bottom: The training data, $+$, test data, $*$, and the ANN output function at different stages of training. The number above each curve corresponds to the numbered points in the top figure.

5 The Problem of Overfitting

One way of trying to find the optimum weights is to simply minimize, using back-propagation with respect to the weights, the MSE,

$$\frac{1}{M}(h(\mathbf{t}) - d(\mathbf{t}))^T(h(\mathbf{t}) - d(\mathbf{t})),$$

where $\mathbf{t}$ is a vector of the training data ages, d is the true growth function, M is the number of training points and h is giving by Equation 1. Figure 4 illustrates how the training and test errors behave as training proceeds. It should be noted that the numerical value of the MSE is a linear mapping of the true MSE in this case. The reason for this is that the training was done on a linear map of the actual data.

The top part of the figure shows the MSE of the training and test data. The test data is the last five data points that have in no way been used in the training process. As can be seen, the training MSE decreases rapidly at first and then levels out but continues to get closer to zero as the iterations increase. While the test MSE reaches a minimum and then starts increasing again. Note that we refer to the second minimum at about thirty five-thousand iterations, the first minimum corresponds to a training MSE that is too high to be valid. The bottom part of the figure shows the actual data points and the ANN output corresponding to the different stages in the top figure. The $+$'s are used to represent the training data and the $*$'s are used to represent the test data. The first ANN output labeled 1 is the output at the stage in training denoted by a 1 in the top part of the figure. As can be seen, the training error is quite low but the test error is still high. At 2 the test error is optimum. At 3 the training error has continued to decrease but the test error is high again. So we can conclude that it is not desirable to just reduce the training error to as small a value as possible. Optimally we would like to stop training as

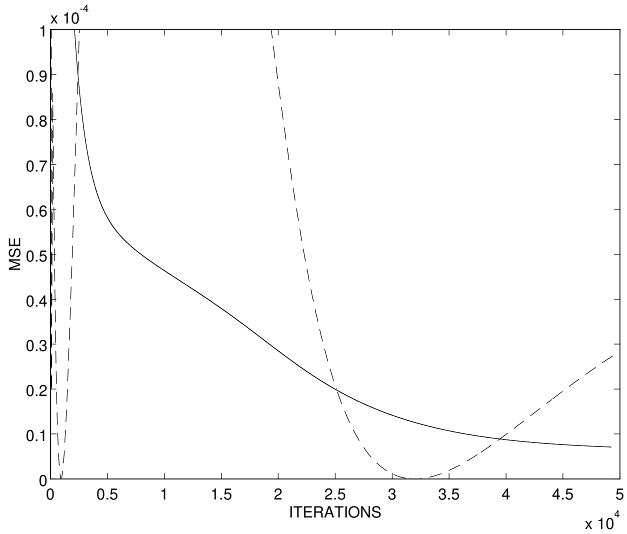

Figure 5: The training MSE (solid line) and the cross validation MSE (dotted) plotted against the number of iterations.

close to the point 2 as possible. One way of explaining this phenomenon is that the ANN has seven parameters and there are only fifteen points used to determine those parameters. Thus the data can easily be overfitted. The data does contain some measurement and external factor errors and these can have an undue influence on the ANN function if the training MSE is reduced by too much. Thus, some way of deciding when to stop training in order to optimize the ANNs prediction abilities is needed.

6 Cross-Validation

A common statistical procedure to avoid overfitting is cross-validation, [6]. The training data is split into two categories. The one category is used to modify the weights in the usual way. The other category is used to cross-validate the data, ie the performance of the ANN on this data is monitored and when it reaches an optimum point, training is stopped.

Since only fifteen data points were to be used for training, only one of them was used for cross-validation so as not to unduly reduce the already very small training set. The last training data point was used as the cross-validation point, since it was closest to the test points.

Figure 5 shows how the MSE of the training data and cross-validation data change as the iterations increase.

As can be seen the cross-validation MSE reaches a minimum at just over thirty thousand iterations and then starts to increase again. Training is stopped at this minimum.

However, with such a small amount of data we can ill afford to even sacrifice one

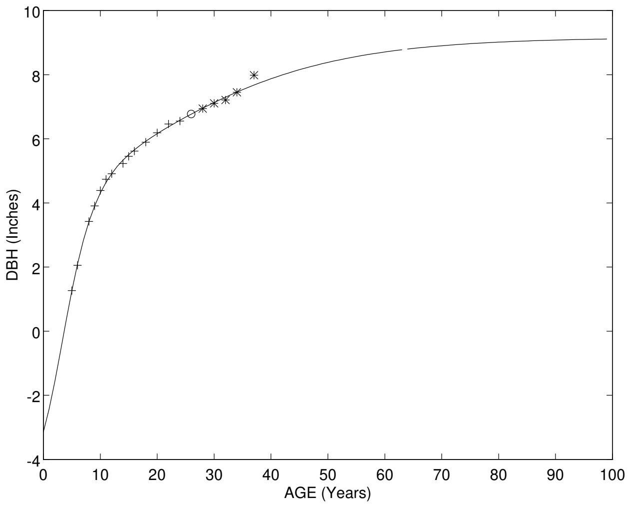

Figure 6: ANN prediction of plot1 DBH. The + denote the training data, the circle denotes the cross-validation data and the $*$ denotes the test data.

training data point. A way around this is to first train the data with the one data point left out as a cross-validation point. Then note the training MSE when the cross-validation MSE is a minimum. This MSE is the lower bound on the MSE before overfitting begins to occur. Next, the ANN is retrained, starting with the same initial values for the weights. This time all the training data is used. Training is stopped when the previously evaluated lower bound on the training MSE is reached.

7 Results of ANN Prediction.

This ANN scheme was applied to the data in Figure 1. The first fourteen points were used in a back-propagation algorithm to modify the weights and the fifteenth point was used as a cross-validation point. The result is shown in Figure 6.

As can be seen, the training MSE is low and the test MSE is also low. The ANN has successfully predicted the next four years of data. The fifth year of future data is a bit out but this is not unexpected since no external factors, such as density of trees in the plot or annual rainfall, have been explicitly taken into account. Certainly the ANN prediction is considerably better than the Schnutte function prediction that was displayed in Figure 2.

This ANN scheme was applied to ten different plots. The first fourteen data points were used in finding the optimal weights, the fifteenth data point was used as a cross-validation point. Retraining was done using all fifteen points and stopped when the lower threshold for the MSE, previously determined, was reached. The remaining points were then used to test how well the ANN model extrapolated.

The results in most cases were good. A sample of the different results which were achieved will now be looked at. In a few cases the results were very good. Figure 7 shows one such case. Here as can be seen a very low MSE is achieved for the test

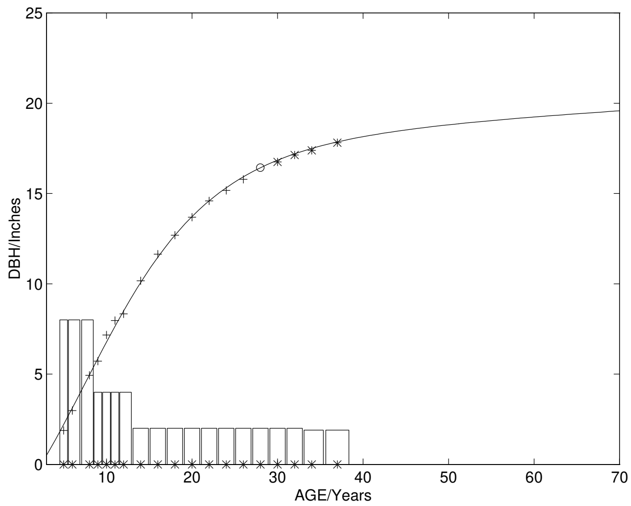

Figure 7: ANN prediction of plot12 DBH. The bars enclosing each data point are proportional in size to the number of trees in the unit area at that time. The first bar corresponds to 80 trees and the last bar corresponds to 19 trees. The training MSE was 0.037, the test MSE was 0.0066. The mean error (ME) gives a measure of the bias. The training ME was 0.007 and the test ME was 0.070.

values.

An interesting case is shown in Figure 8. As can be seen the eighth data point is lower than the preceding data points. The reason for this is at this stage of the thinning the unusual choice of removing the larger trees instead of the smaller trees was made. This caused the average DBH to decrease. As can be seen, due to the reduced competition, the growth function quickly returned to its old trend. It might be thought that such an anomaly would reduce the effectiveness of the ANNs extrapolation ability. But as can be seen it has had little effect. This is an illustration of the well known robust character of ANN models.

Unfortunately, this approach was not successful for all the plots. There were two cases for which satisfactory results were not obtained. The first case is illustrated in Figure 9. As can be seen the extrapolation is quite poor in this case. Examining the data this is perhaps unsurprising. As can be seen, the test data does appear some what anomalous. Particularly the second from last point which hardly shows any growth at all. This is probably due to some other external factors which have not been accounted for.

The other unsuccessful case, seemed of a different nature. It is shown in Figure 10. Unlike the previous case the data does not appear to have a anomalous test set. Yet there is a marked bias in the extrapolation. Perhaps the reason for this might be the fact that the training MSE is still high, i.e. 0.1828 as compared to 0.0368 in the successful case illustrated in Figure 7. This indicates that the cross-validation procedure might have been too conservative in its estimate of the lower bound for the training MSE in order to avoid overfitting. One possible remedy might be to use a more elaborate cross-validation procedure, e.g. the Bootstrap, [1].

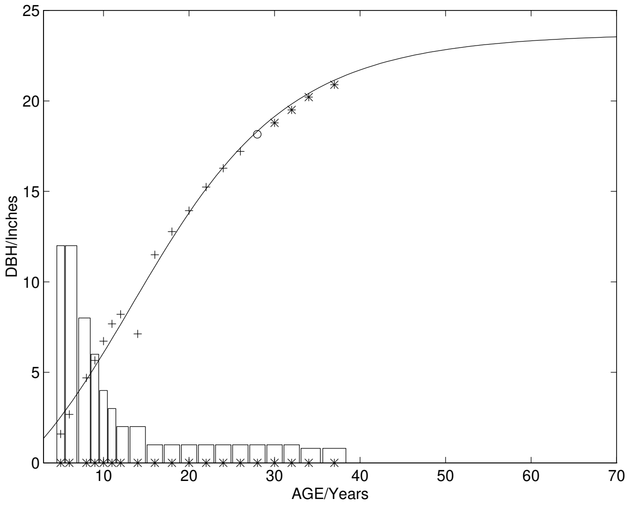

Figure 8: ANN prediction of plot8 DBH. The first bar corresponds to 120 trees and the last bar corresponds to 8 trees. The training MSE was 0.561, the test MSE was 0.083.

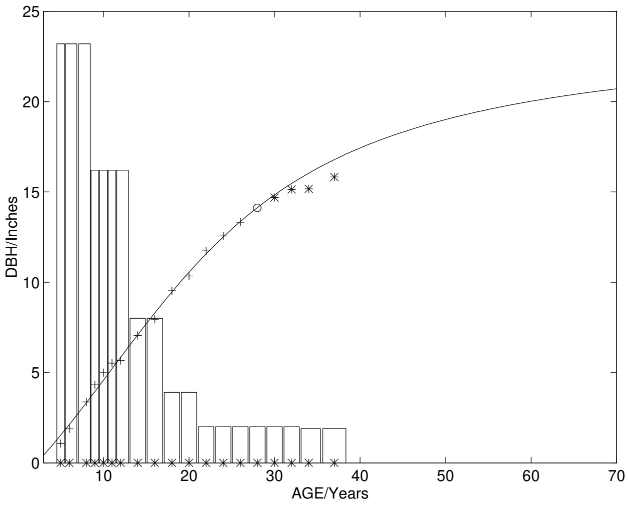

Figure 9: ANN prediction of plot10 DBH. The first bar corresponds to 232 trees and the last bar corresponds to 19 trees. The training MSE was 0.0657, the test MSE was 0.454.

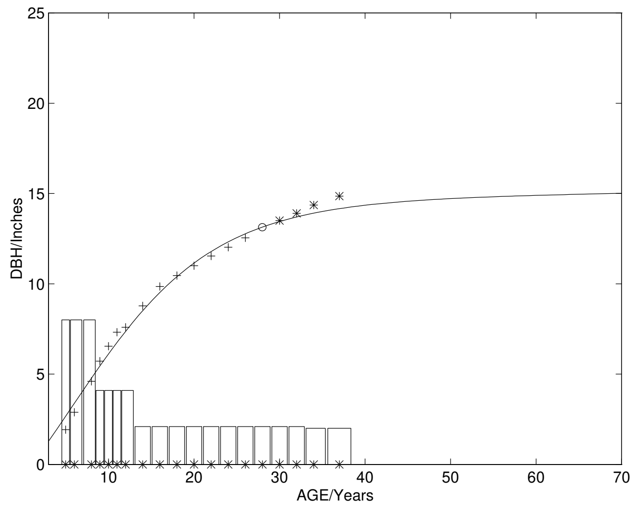

Figure 10: ANN prediction of plot18 DBH. The first bar corresponds to 80 trees and the last bar corresponds to 20 trees. The training MSE was 0.132, the test MSE was 0.183.

However, most of the cases gave fairly good results. A typical example is illustrated in Figure 11.

8 Conclusion

In this paper we have shown how ANNs can be successfully used to model forest tree growth. It was shown that ANNs can perform markedly better than a traditional approach to this problem.

The main theme of the paper has been on the necessity to avoid overfitting the data. It has been illustrated how this can be achieved by the use of cross-validation.

However, it should be noted that when only interpolation between data points was needed overfitting did not seem to present such a problem. This can be seen in Figure 4. The predicted values of the ANN in between data points did not change by much even when overfitting occurred. It is natural to expect that extrapolation is much more sensitive to the model used. For interpolation all that is required is that the model provide a smooth fit between the training data points. If too many hidden layer neurons were used it is possible that the ANN would no longer do this. But when the number of hidden layer neurons is kept low the ANN does not have enough degrees of freedom to provide a fit which is not smooth.

In conclusion, cross-validation can provide a valuable technique for avoiding overfitting in the ANN training process. This is likely to be especially important when extrapolation of the training data is required.

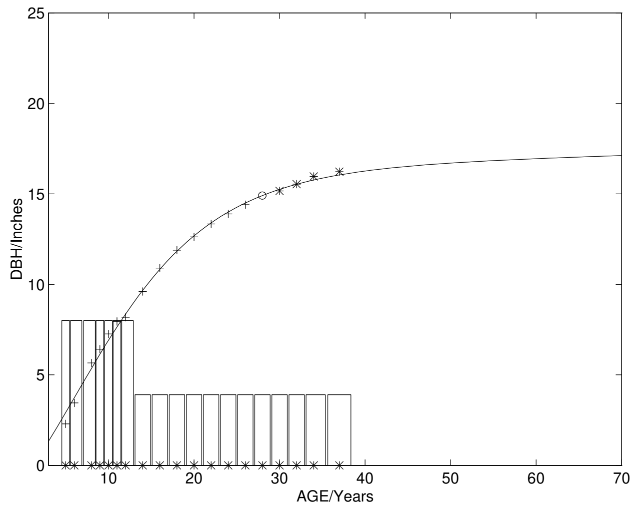

Figure 11: ANN prediction of plot16 DBH. The first bar corresponds to 80 trees and the last bar corresponds to 39 trees. The training MSE was 0.066, the test MSE was 0.020.

References

[1] Efron, B. and R. J. Tibshirani, 1993. *An Introduction to the Bootstrap.* New York: Chapman & Hall

[2] Falkenhagen, E. 1994. *Private Correspondence.*

[3] Haykin, S. 1994. *Neural Networks: A Comprehensive Foundation.* New York: Macmillan College Publishing Company.

[4] Hornik, K. M. Stichcombe, and H. White, 1990. "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks." *Neural Networks* **3**, 551–560.

[5] MacKay, D. J. C. 1991. "Bayesian Interpolation." In [8].

[6] Perrone, M. P., 1993 *Improving Regression Estimation: Averaging Methods for Variance Reduction with Extensions to General Convex Measure Optimization.* PhD (Physics) Thesis at Brown University.

[7] Ratkowsky, D. A. 1990. *Handbook of Nonlinear Regression Models.* New York: Marcel, Dekker, Inc.

[8] Smith C. R. et al. (eds.), *Maximum Entropy and Bayesian Methods*, Seattle, 1991, Kluer Academic Publishers. Netherlands.

[9] Vanclay, J. K., 1994. "Simulation models for tropical forests." *Still to be Published.*