Optimizing the number of hidden nodes of a feedforward artificial neural network

Conference Paper · June 1998

DOI: 10.1109/IJCNN.1998.686018 · Source: IEEE Xplore

Lizelle Fletcher
University of Pretoria
94 PUBLICATIONS **445** CITATIONS
SEE PROFILE

Francois E Steffens
University of Pretoria
46 PUBLICATIONS **580** CITATIONS
SEE PROFILE

Andries Engelbrecht
University of Pretoria
300 PUBLICATIONS **13,465** CITATIONS
SEE PROFILE

Optimizing the Number of Hidden Nodes of a Feedforward Artificial Neural Network

L. Fletcher, V. Katkovnik, F.E. Steffens, University of South Africa, Pretoria, South Africa
A.P. Engelbrecht, University of Pretoria, Pretoria, South Africa

Abstract— **The number of hidden nodes is a crucial parameter of a feedforward artificial neural network. A neural network with too many nodes may overfit the data, causing poor generalization on data not used for training, while too few hidden units underfit the model, and is not sufficiently accurate. The mean square error between the estimated network and a target function has a minimum with respect to the number of nodes in the hidden layer, and is used to measure the accuracy. In this paper an algorithm is developed which optimizes the number of hidden nodes by minimizing the mean square error over noisy training data. The algorithm combines training sessions with statistical analyses and experimental design to generate new sessions. Simulations show that the developed algorithm requires fewer sessions to establish the optimal number of hidden nodes than using the straightforward way of eliminating nodes successively one by one.**

Keywords— **Feedforward Artificial Neural Networks, Nonlinear Regression, Optimizing the Number of Hidden Nodes, Singular Value Decomposition, Likelihood-ratio Test**

I. Introduction and problem setting

Define a univariate response nonlinear model given by noisy data as

$$\mathbf{t}(\mathbf{x}_p) = g(\mathbf{x}_p) + \mathbf{e}(\mathbf{x}_p) \text{ with } p = 1, 2, ..., P \qquad (1)$$

where the data $\mathbf{x}_p \epsilon \Re^I$ and $\mathbf{e}(\mathbf{x}_p) \overset{IID}{\sim} N(0, \sigma^2)$.

As it has been shown ([5], [7]) that any continuous mapping can be accurately approximated by a single hidden layer, the unknown nonlinear function $g(\mathbf{x}_p)$ is fitted to the data $(\mathbf{t}(\mathbf{x}_p), \mathbf{x}_p)$ over all $p = 1, 2, ..., P$ by a one hidden layer, feedforward artificial neural network (ANN) with H hidden nodes. In matrix notation the model is expressed as

$$\mathbf{z}(\mathbf{x}_p, \mathbf{v}_H, \mathbf{w}_H) = \varphi(\mathbf{y}(\mathbf{x}_p, \mathbf{v}_H, \mathbf{w}_H)) \qquad (2)$$

with individual elements

$$z_p = \varphi\left(\sum_{h=1}^{H} w_h y_{hp} + w_0\right)$$

where

$$y_{hp} = \phi(\sum_{i=1}^{I} v_{hi} x_{ip} + v_{0h})$$

with I the total number of input nodes. The total number of parameters (weights) to be estimated is $H(I+1) + (H+1)$.

Here $\mathbf{z} = (z_p){:}P \times 1$ is the output vector of the ANN, $\mathbf{X} = (x_{ip})\,{:}I \times P$ is the matrix of input variables, $\mathbf{w}_H$ is the $(H+1)$-dimensional weight vector between the hidden layer and the output layer, $\mathbf{v}_H = (v_{hi})$ is the $H(I+1)$ weight matrix where v_{hi} is the weight connecting the i-th input variable to node h of the hidden layer, ϕ is a sigmoid function and φ is either a linear or a sigmoid function.

In what follows the mean square error (MSE) risk function is used to characterize the accuracy of the model, defined by

$$\mathbf{R}_{P,H}(\mathbf{v}_H, \mathbf{w}_H) = \frac{1}{P}\sum_{\mathbf{x}_p} \mathbf{E}(g(\mathbf{x}_p) - \mathbf{z}(\mathbf{x}_p, \widehat{\mathbf{v}}_H, \widehat{\mathbf{w}}_H))^2 \qquad (3)$$

where $\widehat{\mathbf{v}}_H$ and $\widehat{\mathbf{w}}_H$ are estimates of $\mathbf{v}_H$ and $\mathbf{w}_H$ respectively and $\mathbf{E}$ denotes the expectation.

In terms of approximation theory and statistics, fitting the data using model (2) is a parametric nonlinear regression problem specified by the structure of the model and the sigmoid function used in it ([2], [3], [10]). From the many theoretical results concerning this problem we wish to note only a few which are of importance for the problem set out in this paper. We specifically emphasize that, from an accuracy analysis, the optimal number of nodes can be found for the problem with additive noise.

Let $\widehat{\mathbf{v}}_H, \widehat{\mathbf{w}}_H$ be the set of weights of the ANN found as a solution in a standard training session to the least squares problem

$$\mathbf{Q}_{P,H}(\mathbf{v}_H, \mathbf{w}_H) = \frac{1}{P}\sum_{\mathbf{x}_p}(\mathbf{t}(\mathbf{x}_p) - \mathbf{z}(\mathbf{x}_p, \mathbf{v}_H, \mathbf{w}_H))^2 \quad (4)$$

i.e. $\mathbf{Q}_{P,H}$ is an empirical risk and

$$(\widehat{\mathbf{v}}_H, \widehat{\mathbf{w}}_H) = \arg\min_{\mathbf{v}_H, \mathbf{w}_H} \mathbf{Q}_{P,H}(\mathbf{v}_H, \mathbf{w}_H). \qquad (5)$$

0-7803-4859-1/98 $10.00©1998 IEEE

Then asymptotically, as $P \to \infty$, subject to some unrestrictive assumptions about $g(\mathbf{x}_p)$,

$$\mathbf{R}^*_{P,H} = \mathbf{R}_{P,H}(\widehat{\mathbf{v}}_H, \widehat{\mathbf{w}}_H) \approx \frac{c_1}{H} + c_2 \frac{H}{P} \log(P) \qquad (6)$$

where c_1, c_2 are constants [2].

Minimizing function (6) with respect to H gives the optimal number of nodes

$$H^* = \left(\frac{c_1}{c_2} \cdot \frac{P}{\log P} \right)^{\frac{1}{2}} . \qquad (7)$$

The first term on the right-hand side of (6) corresponds to the approximation (bias) error of fitting the ANN and the second one to the variance of the random errors. Hence the optimal number of nodes in (7) represents a bias-variance trade-off. Overfitting, i.e. $H > H^*$, thus not only involves extra computation but is also undesirable from the point of view of the accuracy achieved: generalization performance is degraded in terms of ANN theory. Note that as the number of observations(patterns) P increases, the optimum number of nodes H^* also increases proportionally. Optimization of ANNs based on the accuracy criteria set out in (3) and (6) is the primary objective of this paper.

The empirical risk $\mathbf{Q}_{P,H}(\widehat{\mathbf{v}}_H, \widehat{\mathbf{w}}_H)$ is a monotonically decreasing function of H. It therefore does not have a minimum with respect to H and cannot be used instead of $\mathbf{R}^*_{P,H}$ in (6). This situation is quite typical of statistical problems dealing with model selection and in particular with selection of the complexity of the model.

A great amount of effort has been expended to find good approximations for $\mathbf{R}_{P,H}(\mathbf{v}_H, \mathbf{w}_H)$ and (4) - (7) in the form

$$\widehat{H} = \arg\min_H (\log \mathbf{Q}_{P,H}(\widehat{\mathbf{v}}_H, \widehat{\mathbf{w}}_H) + \psi(\widehat{\mathbf{v}}_H, \widehat{\mathbf{w}}_H)) \qquad (8)$$

where $\psi(\widehat{\mathbf{v}}_H, \widehat{\mathbf{w}}_H)$ is a penalty function increasing with the complexity (i.e. number of nodes) of the model (see e.g. [6], [8], [11]) .

It has been shown that for linear regression models, methods such as cross-validation, generalized cross-validation, Akaike and the C_p criteria differ only by the penalty function $\psi(\widehat{\mathbf{v}}_H, \widehat{\mathbf{w}}_H)$ in (8) (e.g. [4], [8], [11], [12]). Procedure (8) is computationally efficient, but the function $\psi(\widehat{\mathbf{v}}_H, \widehat{\mathbf{w}}_H)$ and the approach on the whole are well justified only for linear regression models. Loss functions having features of (8) have been developed specifically for ANNs ([2], [10]). The proposed penalty functions include uncertain parameters and functions which enable (8) to have a minimum on H. However the practical applications of these methods are questionable due to the above-mentioned ambiguity embedded in $\psi(\widehat{\mathbf{v}}_H, \widehat{\mathbf{w}}_H)$.

Cross-validation methods (e.g. [1], [6]) in their original combinatorial form are in many cases able to provide good approximations to problem (6) but are computationally intensive.

The "one-out" cross-validation loss function has the form

$$\mathbf{Q}^{[CV]}_{P,H} = \frac{1}{P_C} \sum_{\mathbf{x}_p \epsilon \mathbf{X}_C} (\mathbf{t}(\mathbf{x}_p) - \widehat{\mathbf{z}}(\mathbf{x}^{[p]}, \mathbf{v}^{[p]}_H, \mathbf{w}^{[p]}_H))^2 \qquad (9)$$

where

$$\begin{aligned} & (\mathbf{v}^{[p]}_H, \mathbf{w}^{[p]}_H) & (10) \\ = \; & \arg \min_{\mathbf{v}_H, \mathbf{w}_H} \frac{1}{P-1} \sum_{r \neq p} (\mathbf{t}(\mathbf{x}_r) - \widehat{\mathbf{z}}(\mathbf{x}_r, \mathbf{v}_H, \mathbf{w}_H))^2 & \end{aligned}$$

$\mathbf{X}_C$ is a control set, P_C is the number of observations in it, and a permutation over a set of observations is assumed in (9) - (10).

A simplified form of cross-validation where the observations are split into two sets - one for learning and one for control – is usually quite practical and good results are obtained when the number of observations is much larger than the number of parameters to be estimated.

In this paper we employ a different approach based on a statistical analysis of the results of training sessions. Consider the empirical risk $\mathbf{Q}_{P,H}(\widehat{\mathbf{v}}_H, \widehat{\mathbf{w}}_H)$ in (4) with parameters $\widehat{\mathbf{v}}_H$ and $\widehat{\mathbf{w}}_H$ found from (5) as a function of H. When the model is substantially overfitted the bias components are compensated for, however the sum of random errors becomes large. A statistical test can be used to evaluate $\mathbf{Q}_{P,H}(\mathbf{v}_H, \mathbf{w}_H)$ with decreasing H to find the critical area of $\widehat{H}$ where the bias becomes non-negligible. This critical value $\widehat{H}$ does not actually render the compromise bias-variance in (6) but simply serves to identify an area of H values where further decreasing the number of nodes becomes counter-effective, while an increase does not improve the generalization abilities of the ANN.

Depending on the significance level α used in the statistical tests we obtain an overfitted ($\widehat{H} > H^*$) or underfitted ($\widehat{H} < H^*$) ANN. This statistical test approach can consequently be treated as an approximate solution to (6)-(7).

The main contribution of this paper is the development of a recursive algorithm optimizing the ANN. The algorithm comprises training sessions of the ANN, statistical analyses of the results and experimental design in subsequent sessions. The central idea behind the design is to minimize the number of training sessions necessary for optimization of the number of hidden nodes.

In the experimental design the null hypothesis H_0 : $\mathbf{w}_{\triangle H} = 0$ is tested for output weights in order to establish the number of nodes $\triangle H$ that can supposedly be eliminated from the model. These $\triangle H$ nodes are specified in order to preserve the weight values achieved for the remaining nodes from the previous session. A singular value decomposition (SVD) of the introduced conditional Fisher information matrix of the output weights and partitioned maximum likelihood methods have been developed for an ANN setting.

The algorithm proposed is different in a number of aspects from previously reported results.

Research closely related to our proposal is architecture selection of a neural network as reported by Steppe et.al. [9] where the likelihood-ratio test statistic is used as a model selection criterion to compare ANNs with a decreasing number of nodes. The criterion is used in a sequential procedure where successive models differ by one hidden node. According to the algorithm in [9] the ANN, starting from an initial architecture with large H, requires a full training procedure for each successive model until $\widehat{H}$ is found. In our algorithm successive sessions can be different by a large number of nodes $\triangle H$, resulting in a substantial saving in the amount of training sessions.

Xue et al. [13] reported using the SVD to find the optimal number of hidden nodes of a neural network. Contrary to their method, our algorithm does not base the SVD on the Fisher information matrix of all the estimated parameters, but uses the conditional Fisher information matrix restricted to the output weights only. This decreases the dimensionality of the matrix and produces more readily interpretable results.

II. Optimization algorithm

Basically, the algorithm specifies the conditional statistical analysis of a completed training session to plan the number of nodes for the following training session and to specify which nodes to preserve. The term "conditional" means here that in the statistical analysis the input weights $\mathbf{v}_H = \widehat{\mathbf{v}}_H$ are assumed to be fixed, while the output weights $\mathbf{w}_H$ vary. The multiple hypotheses to be tested have the form

$$H_{0q} : w_k = 0 \quad \forall \quad k \in K_q \tag{11}$$

where K_q, $q = 1,..,Q$, are subsets of nodes considered for elimination from the ANN.

The sets K_q in (11) are formed by analysis of the conditional Fisher information matrix and differ by either the number of nodes or the specific nodes identified for elimination. The usual likelihood-ratio procedure is used to test (11) for any given K_q.

The number of nodes is determined by the nonzero output weights.

The conditional Fisher information matrix ($H \times H$) is introduced in this paper as follows

$$\Phi(\mathbf{v}_H, \mathbf{w}_H) = \frac{1}{P} \sum_{\mathbf{x}_p} \frac{\partial z(\mathbf{x}_p, \widehat{\mathbf{v}}_H, \mathbf{w}_H)}{\partial \mathbf{w}_H} \cdot \frac{\partial z(\mathbf{x}_p, \widehat{\mathbf{v}}_H, \mathbf{w}_H)}{\partial \mathbf{w}_H^T}. \tag{12}$$

The singular value decomposition of (12) is given by

$$\begin{aligned} \Phi &= A \Lambda A^T, \qquad (13) \\ \Lambda &= diag(\lambda_1, ..., \lambda_H) \geq 0 \end{aligned}$$

with λ_H the smallest eigenvalue of Φ, and $\mathbf{a}_H$ the H-th column of matrix A. It is shown that if the absolute values of the elements of $\mathbf{a}_H$ are very different, then the largest of these elements, say a_{H_i} and a_{H_j}, indicate the suspect nodes, say n_i and n_j, that can be excluded from the ANN. This property is developed in order to form the sets K_q corresponding to the groups of the smallest eigenvalues in Λ.

On the whole the algorithm constitutes the following basic steps:

1. **Initialization:**
   Train the ANN with $H = H^0$ hidden nodes.
   Go to Step 2.
2. **Design a training session** in the following two stages:
   (a) Stage I: Perform a singular value decomposition of the conditional Fisher information matrix to determine the sets K_q.
   (b) Stage II: Test hypothesis (11) by the likelihood-ratio statistic

$$L_1 = \left(\frac{SSE_R - SSE_F}{df_R - df_F} \right) \bigg/ \left(\frac{SSE_F}{df_F} \right) \tag{14}$$

where
$df_F = IP - (H+1)$ and SSE_F are the number of degrees of freedom and the corresponding sum of squared errors (SSE) obtained for the training session respectively;
$df_R = IP - (H + 1 - \triangle H_q)$ is the number of degrees of freedom under hypothesis H_{0q} where $\triangle H_q$ is the number of weights in the set K_q, and

$$SSE_R = \min_{\mathbf{w}_{H-\triangle H_q}} \sum_p (t_p - z(\mathbf{x}_p, \widehat{\mathbf{v}}_H, \mathbf{w}_{H-\triangle H_q}))^2. \tag{15}$$

If $L_1 \leq F_{\alpha;df_R-df_F;df_F}$, accept hypothesis H_{0q}, i.e. the reduced ANN is adopted.

Use (14) to search over all sets K_q, $q = 1, ..., Q$ to find the maximum number of nodes that can be eliminated. Denote this number be ΔH^*.

If $\Delta H^* \geq \Delta H_{crit}$ go to Step 3.

If all of the hypotheses H_{0q}, $q = 1, .., Q$ are rejected, i.e. $\Delta H^* = 0$, or $\Delta H^* < \Delta H_{crit}$, the algorithm is stopped and the ANN has H nodes. Here ΔH_{crit} is specified by the researcher, and is determined by the complexity of the model.

3. **Experiment:**
   Train the ANN with $H - \Delta H^*$ nodes.
4. **Analysis** where two successive training sessions are compared using the likelihood-ratio test statistic

$$L_2 = \left(\frac{SSE_R - SSE_F}{df_R - df_F}\right) \Big/ \left(\frac{SSE_F}{df_F}\right) \qquad (16)$$

where df_F and df_R are the degrees of the freedom corresponding to the ANNs, and SSE_F and SSE_R are the corresponding sums of squared residuals.

If $L_2 \leq F_{\alpha;df_R-df_F;df_F}$, the reduced model with $H - \Delta H^*$ nodes is confirmed by the experiment. Go to Step 2 to further decrease the number of nodes.

If $L_2 > F_{\alpha;df_R-df_F;df_F}$, the reduced model with $H - \Delta H^*$ nodes is rejected. Return to Step 2 to reassess the design of the training session. The number of nodes in this case in assumed to be approximately equal to $(\Delta H_{crit} + \Delta H^*)/2$.

The likelihood ratios (14) and (16) are essentially different as the SSE used in (14) is calculated by varying only the output weights and is therefore easy to calculate, while the SSE in (16) involves a training session, i.e. the calculation of all output and input weights.

III. Implementation of the Algorithm and Simulation

The developed algorithm is implemented using the Neural Network Toolbox of Matlab version 5.1.

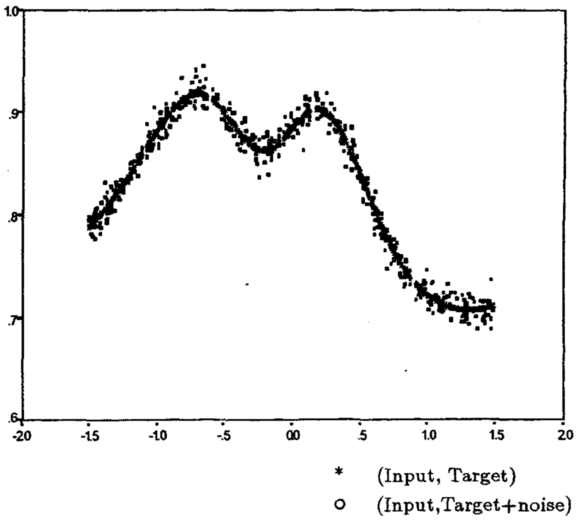
Simulation experiments produced with regressions which are determined as ANNs of which the architecture is known (number of nodes and weights) show that the algorithm is quite efficient and requires only a moderate number of training sessions.

Testing at the $\alpha = 0.05$ significance level provides the most accurate results while results with $\alpha = 0.1$ and $\alpha = 0.01$ results in oversmoothing and undersmoothing respectively.

For example, ANN regression models with $H^* = 5$ hidden nodes and initialization $H^0 = 30$ nodes reach their goals within 5 to 8 training sessions with accuracy between 1 and 3 nodes.

As a particular result, consider the data presented in Figure 1. The 500 input patterns are plotted on the X-axis vs. the corresponding target values on the Y-Axis. Noise was generated from a $N(0, (0.05)^2)$ distribution and added to the target values. These targets with the additive noise are also plotted on the Y-axis of Figure 1. This vector association problem corresponds to an ANN regression with $H^* = 5$ and $P = 500$. ANNs were trained with $H^0 = 30$ and $\Delta H_{crit} = 1$.

Fig. 1. The vector association problem



Figures 2 - 4 display the results of the ANNs obtained from implementation of the algorithm for $\alpha = 0.1$, $\alpha = 0.05$ and $\alpha = 0.01$ respectively, with input values plotted on the X-axis vs. corresponding ANN output values plotted on the Y-axis.

In all cases the number of training sessions is very small compared to the 25 training sessions that would have been necessary if nodes were eliminated successively one by one. With $\alpha = 0.05$ the accurate ANN model with $H = H^* = 5$ hidden nodes was obtained within 6 training sessions. Setting $\alpha = 0.01$ required 8 training sessions and resulting in an ANN with $H = 3$ nodes, while 5 training sessions were required for $\alpha = 0.1$, resulting in an ANN with $H = 8$ nodes.

IV. Summary

A recursive algorithm is developed to optimize the number of hidden nodes in a feedforward ANN. A nonlinear regression statistical model setting is used to

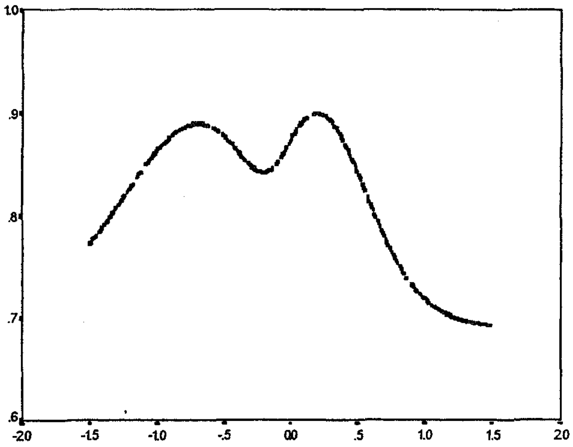


Fig. 2. ANN using algortihm with $\alpha = 0.1$

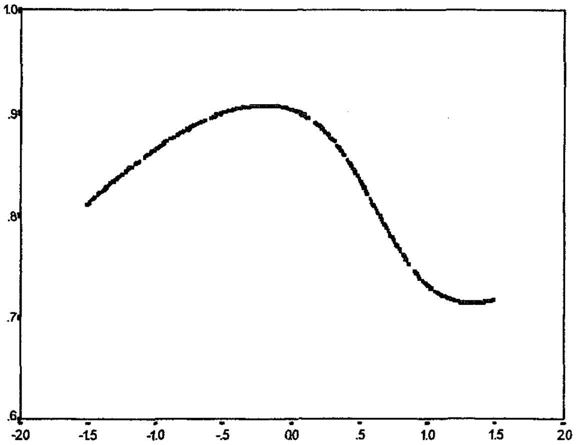


Fig. 4. ANN using algorithm with $\alpha = 0.01$

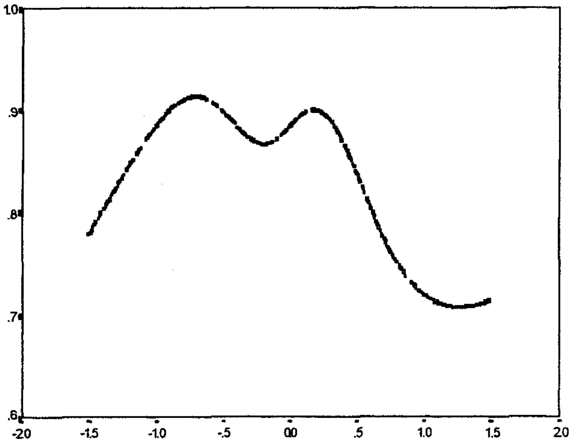


Fig. 3. ANN using algorithm with $\alpha = 0.05$

construct the algorithm. The results of a completed training session of an ANN is statistically analyzed to determine and specify the number of hidden nodes that can be eliminated from the ANN for the design of the subsequent ANN. The procedure is based on a singular value decomposition of the conditional information matrix, and uses likelihood-ratio test statistics as selection criteria for the specific nodes to be eliminated, as well as for the selection of the correct ANN model. A simulation study illustrates the application and effectiveness of the algorithm.

A special modification of the algorithm was developed and applied to radar-rainfall data sets. The aim is to obtain reliable ground rainfall estimates based on radar measurements. Depending on the data sets used, the models obtained consist of 12 - 23 nodes and demonstrate good generalization abilities. The algorithm was initiated with $H^0 = 40$ and took not more than 10 training sessions to reach the final results.

References

[1] S. Amari†, N. Murata, K. R. Müller, M. Finke, and H. Yang. Asymptotic Statistical Theory of Overtraining and Cross-Validation. Technical Report METR95-06, †University of Tokyo, Department of Mathematical Engineering and Information, Physics, University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo 113, Japan, Aug. 1995.

[2] A. R. Barron. Approximation and Estimation Bounds for Artificial Neural Networks. *Machine Learning*, (14):115–133, 1994.

[3] B. Cheng and D. M. Titterington. Neural Networks: A Review from a Statistical Perspective. *Statistical Science*, 9(1):2–54, 1994.

[4] P. R. Cohen. *Empirical Methods for Artificial Intelligence.* MIT Press, London, 1995.

[5] G. Cybenko. Continuous valued neural networks with two hidden layers are sufficient. Technical report, Tufts University, Department of Computer Science, Medford, MA, Mar. 1988.

[6] M. Hassoun. *Fundamentals of Artificial Neural Networks.* MIT Press, 1995.

[7] R. Hecht-Nielsen. Theory of the backpropagation neural network. In *Proc. International Joint Conference on Neural Networks, I*, pages 593–611, New York, June 1989. IEEE Press.

[8] C. Hurrich and C. Tsai. Regression and time series model selection in small samples. *Biometrika*, (76):297–307, 1989.

[9] J. M. Steppe, K. W. Bauer, and S. K. Rogers. Integrated Feature and Architecture Selection. *IEEE Transactions on Neural Networks*, 7(4):1007–1014, July 1996.

[10] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, Inc., 1995.

[11] C. Wei. On predictive least squares principles. *The Annals of Statistics*, 20(1):1–42, 1992.

[12] H. White. *Artificial Neural Networks.* Blackwell, Cambridge, 1992.

[13] Q. Xue, Y. Hu, and W. J. Tompkins. Analyses of the Hidden Units of Back-Propagation Model by Singular Value Decomposition (SVD). *IJCNN*, pages I–739–I–742, 1990.