This article was downloaded by: [ ]
On: 22 September 2011, At: 03:16
Publisher: Taylor & Francis
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK

Connection Science
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/ccos20

Bootstrapping with Noise: An Effective Regularization Technique

YUVAL RAVIV [a] & NATHAN INTRATOR [a]

[a] School of Mathematical Sciences, Sackler Faculty of Exact Sciences, Tel-Aviv University, Ramat Aviv, 69978, Israel

Available online: 01 Jul 2010

To cite this article: YUVAL RAVIV & NATHAN INTRATOR (1996): Bootstrapping with Noise: An Effective Regularization Technique, Connection Science, 8:3-4, 355-372

To link to this article: http://dx.doi.org/10.1080/095400996116811

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: http://www.tandfonline.com/page/terms-and-conditions

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan, sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

Downloaded by [ ] at 03:16 22 September 2011

Connection Science, Vol. 8, Nos 3 & 4, 1996, 355–372

Bootstrapping with Noise: An Effective Regularization Technique

YUVAL RAVIV & NATHAN INTRATOR

Bootstrap samples with noise are shown to be an effective smoothness and capacity control technique for training feedforward networks and for other statistical methods such as generalized additive models. It is shown that noisy bootstrap performs best in conjunction with weight-decay regularization and ensemble averaging. The two-spiral problem, a highly non-linear, noise-free data, is used to demonstrate these findings. The combination of noisy bootstrap and ensemble averaging is also shown useful for generalized additive modelling, and is also demonstrated on the well-known Cleveland heart data.

KEYWORDS: Noise injection, combining estimators, pattern classification, two-spiral problem, clinical data analysis.

Downloaded by [ ] at 03:16 22 September 2011

1. Introduction

The bootstrap technique has become one of the major tools for producing empirical confidence intervals of estimated parameters or predictors (Efron & Tibshirani, 1993). One way to view bootstrap is as a method to simulate noise inherent in the data, and thus increase effectively the number of training patterns. A simple bootstrap procedure amounts to sampling with return from the training data, and constructing several training sets, all with the same size as the original training set. Later, the variability between the estimated parameters can be measured, and give some indication about the true variability of the model parameters arising from the data. Furthermore, variability of the prediction, or error bars on the prediction, can also be estimated in this way.

One variant of bootstrap involves estimation of a model of the form

$$y = f(x) + \varepsilon$$

for some parametric family to which f belongs, and a noise ε which is assumed to be small with zero mean. Once an empirical function $\hat{f}$ has been estimated from n training samples, there remains a noise vector $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)$. One can then sample from the empirical distribution of the noise by sampling (with return) from ε_i and constructing new samples of the form $(x_i^\star, y_i^\star)$, in which ε_i was replaced by $\varepsilon_i^\star$ sampled from the above set. Clearly, this approach can be easily extended to a

Y. Raviv and N. Intrator, School of Mathematical Sciences, Sackler Faculty of Exact Sciences, Tel-Aviv University, Ramat Aviv 69978, Israel. E-mail: {yuv,nin}@math.tau.ac.il. Present address of N. Intrator, Institute of Brain and Neural Systems, Box 1843, Brown University, Providence, RI 02912, USA.

0954–0091/96/030355–18 $7.50 © 1996 Journals Oxford Ltd

smoothed bootstrap (Efron & Tibshirani, 1993) by samping from the empirical distribution by ε_i rather than just sampling from the set of ε_i's. In such case, one can increase the size of each boostrap set, since due to the noise, the different sets are sufficiently independent. It should be noted that if $\hat{f}$ is biased, the noise vector may be over-estimated.

For classification problems, the form

$$y = f(x + \varepsilon)$$

may be more appropriate. In this case, using noise injection to the inputs during training can improve the generalization properties of the estimator (Sietsma & Dow, 1991). Recently, Bishop (1995) has shown that training with small amounts of noise is locally equivalent to smoothness regularization. In this paper, we give a different interpretation to noise added to the input during training, and view it as a regularizing parameter that controls, in conjunction with ensemble averaging, the capacity and the smoothness of the estimator. The major role of this noise is to push different estimators to different local minima, and so produce a more independent set of estimators. Best performance is then achieved by averaging over the estimators. For this regularization, the level of the noise may be larger than the 'true' level which can be indirectly estimated. Since we want to study the effect of bootstrapping with noise on the smoothness of the estimator, separated from the task of input noise estimation, we consider a highly non-linear, noise-free classification problem, and show that even in this extreme case, addition of noise during training improves results significantly.

We chose a problem that is very difficult for feedforward neural networks (NNs). It is difficult due to the highly non-linear nature of the decision boundaries, and the fact that these non-linearities are easier to represent in local radially symmetric functions rather than in ridge functions such as those given by feedforward sigmoidal functions. Since the training data are given with no noise, it seems unreasonable to train a network with noise, but we show that even in this case training with noise is a very effective approach for smoothing the estimator. In addition to demonstrating our method on a different class of predictors—the generalized additive models—we also apply it to another well-known data set—the Cleveland heart data (Detrano *et al.*, 1989).

Downloaded by [ ] at 03:16 22 September 2011

2. Theoretical Considerations

There are a number of factors that have to be applied carefully when trying to regularize an estimator. The regularization is aimed at finding an optimal trade-off between the variance and bias of the estimator (Geman *et al.*, 1992), and for best performance one has to utilize this decomposition of the error function. The motivation to our approach follows from a key observation regarding the bias variance decomposition, namely the fact that ensemble averaging does not affect the bias portion of the error, but reduces the variance when the estimators on which averaging is done are independent.

2.1. Bias/Variance Trade-off for Ensemble of Predictors

The classification problem is to estimate a function $f_{\mathscr{D}}(x)$ of observed data characteristics x, predicting class label y, based on a given training set $\mathscr{D} = \{(x_1, y_1)\}, \ldots, (X_L, Y_L)\}$ using some measure of the estimation error on $\mathscr{D}$.

A good estimator will perform well not only on the training set, but also on new validation sets which were not used during estimation.

Evaluation of the performance of the estimator is commonly done via the mean squared error (MSE) distance by taking the expectation with respect to the (unknown) probability distribution P of y:

$$E[(y - f_{\mathscr{D}}(x))^2|x, \mathscr{D}]$$

This can be decomposed into

$$E[(y - f_{\mathscr{D}}(x))^2|x, \mathscr{D}] = E[(y|x])^2|x, \mathscr{D}] + E[(f_{\mathscr{D}}(x) - E[y|x])^2]$$

The first term does not depend on the training data $\mathscr{D}$ or on the estimator $f_{\mathscr{D}}(x)$; it measures the amount of noise or variability of y given x. Hence, f can be evaluated using

$$E[(f_{\mathscr{D}}(x) - E[y|])^2]$$

The empirical MSE of f is given by

$$E_{\mathscr{D}}[(f_{\mathscr{D}}(x) - E[y|x])^2]$$

where $E_{\mathscr{D}}$ represents expectation with respect to all possible training sets $\mathscr{D}$ of fixed size.

To see further the performance under MSE, we decompose the error to bias and variance components to get

$$E_{\mathscr{D}}[(f_{\mathscr{D}}(x) - E[y|x)^2] = (E_{\mathscr{D}}[f_{\mathscr{D}}(x)] - E[y|x])^2 + E_{\mathscr{D}}[(f_{\mathscr{D}}(x) - E_{\mathscr{D}}[f_D(x)])^2] \qquad (1)$$

The first term on the right-hand side is called the bias of the estimator and the second term is called the variance. When training on a fixed training set $\mathscr{D}$, reducing the bias with respect to this set may increase the variance of the estimator and contribute to poor generalization performance. This is known as the trade-off between variance and bias. Typically, variance is reduced by smoothing; however, this may introduce bias (since, for example, it may blur sharp peaks). Bias is reduced by prior knowledge. When prior knowledge is used also for smoothing, it is likely to reduce the overall MSE of the estimator.

When training NNs, the variance arises from two terms. The first term comes from inherent data randomness and the second term comes from the non-identifiability of the model, namely, the fact that for a given training data, there may be several (local) minima of the error surface.[1]

Consider the ensemble average $\bar{f}$ of several predictors, e.g. NNs with different random initial weights which are trained on data with added Gaussian noise:

$$\bar{f}(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x)$$

These predictors are identically distributed and, thus, the variance contribution (equation (1)) becomes (we omit x and $\mathscr{D}$ for clarity)

$$\begin{aligned} E[(\bar{f} - E[\bar{f}])^2] &= E\left[\left(\frac{1}{N}\sum f_i - E\left[\frac{1}{N}\sum f_i\right]\right)^2\right] \\ &= E\left[\left(\frac{1}{N}\sum f_i\right)^2\right] + \left(E\left[\frac{1}{N}\sum f_i\right]\right)^2 - 2E\left[\frac{1}{N}\sum f_i E\left[\frac{1}{N}\sum f_i\right]\right] \end{aligned}$$

Downloaded by [ ] at 03:16 22 September 2011

$$= E\left[\left(\frac{1}{N}\sum f_i\right)^2\right] - \left(E\left[\frac{1}{N}\sum f_i\right]\right)^2 \tag{2}$$

The first term on the right-hand side can be rewritten as

$$E\left[\left(\frac{1}{N}\sum f_i\right)^2\right] = \frac{1}{N^2}\sum E[f_i^2] + \frac{2}{N^2}\sum_{i<j} E[f_i f_j]$$

and the second term gives

$$\left(E\left[\frac{1}{N}\sum f_i\right]\right)^2 = \frac{1}{N^2}\sum E[f_i] + \frac{2}{N^2}\sum_{i<j} E[f_i]E[f_j]$$

Plugging these equalities in equation (2) gives

$$E[(\bar{f} - E[\bar{f}])^2] = \frac{1}{N^2}\sum \{E[f_i^2] - (E[f_i])^2\} + \frac{2}{N^2}\sum_{i<j} \{E[f_i f_j] - E[f_i]E[f_j]\} \tag{3}$$

If the predictors $\{f_i\}$ are highly correlated, for example if $f_i = f_j = f$ for all i, j, then the above equation becomes

$$\mathrm{Var}(\bar{f}) = \frac{1}{N}\mathrm{Var}(f) + \frac{2}{N^2}\frac{N(N-1)}{2}\mathrm{Var}(f) = \mathrm{Var}(f)$$

namely, there is no reduction in variance[2] in this case. If the predictors are identically distributed and independent, then the second term drops and we are left with

$$\mathrm{Var}(\bar{f}) = \frac{1}{N}\mathrm{Var}(f_i)$$

Note that

$$E[f_i f_j] - E[f_i]E[f_j] = E(\{f_i - E[f_i]\}\{f_j - E[f_j]\})$$

Thus, the notion of independence can be understood as independence of the deviations of each predictor from the expected values of the predictor, which can be replaced (due to linearity) by

$$E(\{f_i - E[\bar{f}]\}\{f_j - E[\bar{f}]\})$$

and is thus interpreted as an independence of the prediction variation around a common mean.

The success of ensemble averaging of NNs in the past (Breiman, 1994; Hansen & Salamon, 1990; Perrone, 1993; Wolpert, 1992) is due to the fact that NNs have in general many local minima, and thus even with the same training set, different local minima are found when starting from different random initial conditions. These different local minima lead to somewhat independent predictors, and thus the averaging can reduce the variance. When a larger set of independent networks is needed, but no more data are available, data reuse methods can help. Bootstrapping (Breiman, 1994) has been very helpful, since by resampling (with return) from the training data, the independence of the training sets is increased, and hence, the independence of the estimators, leading to improved ensemble results. Smoothed bootstrap (Krogh & Hertz, 1992; Ripley, 1996) is potentially more useful since larger sets of independent training samples can be generated. The smoothed bootstrap approach amounts to generating larger data sets by simulating the true noise in the data.

Downloaded by [ ] at 03:16 22 September 2011

Downloaded by [ ] at 03:16 22 September 2011

3. The Bootstrap Ensemble with Noise Algorithm

In the bootstrap ensemble with noise (BEN), we push the idea of noise injection further; we observe that adding noise to the inputs increases the first term on the right-hand side of equation (3), i.e. adds variance to each estimator, but, on the other hand, decreases the contribution of the second term on the right-hand side as it increases the independence between estimators. Instead of using the 'true' noise (estimated from the data) for bootstrap, we seek an optimal noise level which gives the smallest contribution to the error from the sum of the two components of the variance. It is impossible to calculate the optimal variance of the Gaussian noise without knowing f explicitly; therefore, the value of this variance remains a regularization term: a parameter which has to be estimated so as to minimize the total contribution of the variance to the error. Furthermore, since the injection of noise increases the independence between different training sets, we can use bootstrap sets that are larger than the original training set. This does not affect the bias (if the noise is symmetric around zero) but can reduce the variance. Note that the bias contribution to the error is not affected by introducing the ensemble-average estimator due to linearity of expectations.

It follows that the BEN approach has the potential of reducing the contribution of the variance term to the total error. We thus should seek a different trade-off point between the contribution of the variance and the bias. In other words, we are able to use large (unbiased) networks without being affected by the large variance associated with such networks. This observation implies that the estimation of optimal noise levels should not be based on a single estimator performance, but rather based on the ensemble performance. The large variance of each single network in the ensemble can be tempered with a regularization such as weight decay (Krogh & Hertz, 1992; Ripley, 1996), but, again, the estimation of the optimal regularization factor should be done on the ensemble-averaged performance. Breiman (1994) and Ripley (1996) show compelling empirical evidence for the importance of weight decay as a single network stabilizer. Our results confirm this fact under the BEN model.

The BEN Algorithm

- Let $\{(x_i, y_i)\}$ be a set of training patterns for $i = 1, \ldots, N$.
- Let $\varepsilon = \{\varepsilon_1, \ldots, \varepsilon_J\}$.
- Let $\lambda = \{\lambda_1, \ldots, \lambda_I\}$.
- For a noise level ε_j estimate an optimal penalty term for weight decay λ_i:
  - —Fix a size K for the bootstrap sample, such that $K \gg N$ (we used $K = 10N$).
  - —Let $s_1, s_2, \ldots, s_K$ be a set of indices, chosen from a uniform distribution, $s_i \sim U(1, N)$.
  - —For a ε_j, create a noisy bootstrap resample of the training set inputs: $\{x_{s_i} + \zeta_i\}_{i=1,\ldots,K}$ and the corresponding resampled outputs $\{y_{s_i}\}_{i=1,\ldots,K}$ where ζ_i is a vector whose components are $N(0, \varepsilon_j^2)$.
  - —Train several networks with the noisy samples using weight decay $\lambda_1, \ldots, \lambda_I$.
  - —Generate an ensemble average of the set of networks.
  - —Choose via cross-validation or a test set, the optimal weight decay λ.
- Repeat the process for the new choice of noise ε_j until there is no improvement in prediction.

In the simple case, the same noise level is used for each dimension. This is suitable

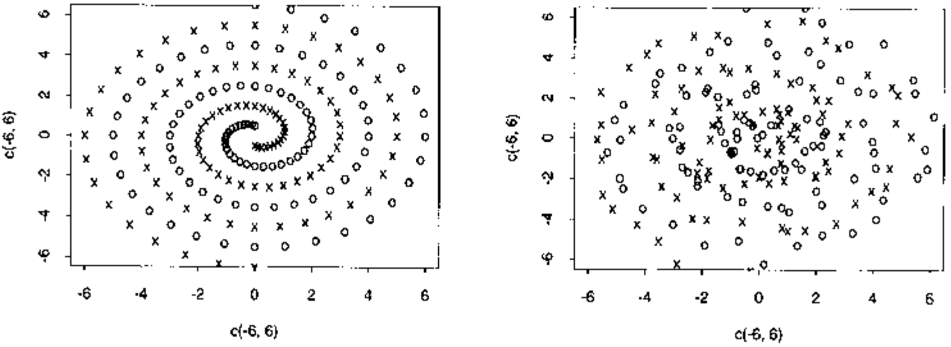


Figure 1. The two-spirals training data (left). Training points with noise—standard deviation, SD = 0.3 (right). As can be seen, the noise level that contaminates the data causes objects to cross the virtual boundary defined by the data, i.e. the noise leads to wrong class labelling for the training data. This reduces performance of single predictors, but the added independence between the predictors leads to improved ensemble performance.

for problems in which each of the dimensions are on the same scale, or, more precisely, when the noise distribution is similar in different data dimensions. When all covariates have the same interpretation, e.g. similar measurements taken at different time steps, or when dealing with pixel data, such noise assumption is adequate; however, when the noise is non-homogeneous in space, has a non-diagonal covariance matrix or when different dimensions represent completely different measurements, it is best to estimate the different noise levels in each dimension separately. When this is too costly, or there is insufficient data for robust estimation, a quick solution is to sphere the data by setting the variance in each dimension to be the same and with zero mean.

3.1. The Two-spirals Problem

The 'two-spirals' problem consists of a training set with 194 X–Y values, half of which are to produce a 1 output and half a 0 output. These training posts are arranged in two interlocking spirals that go around the origin three times, as shown in Figure 1.

The problem was proposed to the CMU benchmark by Alexis Wieland of MITRE Corporation (see Appendix A for a description of the problem). It appears to be extremely hard for backpropogation networks due to its high non-linearity. It is easy to see that the two-dimensional points of the spirals could not be separated by a small combination of linear separators. Lang and Witbrock (1988) proposed a 2–5–5–5–1 network with short-cuts using 138 weights. They used a variant of the quick-prop learning algorithm (Fahlman, 1989) with weight decay. They claimed that the problem could not be solved with simpler architecture (i.e. less layers or without short-cuts). Their result on the same data set seems to give poor generalization results. Baum and Lang (1991) demonstrated that there are many sets of weights that would cause a 2–50–1 network to be consistent with the training set; however, the single-layer feedforward architecture trained with error backpropagation was unable to find any of them when starting with random initial weights.

Downloaded by [ ] at 03:16 22 September 2011

Fahlman and Lebiere (1990) used the cascade-correlation architecture for this problem. They got better results, but still little 'spiralness'. Recently, Deffuant (1995) suggested the 'perceptron membrane' method that uses piecewise linear surfaces as discriminators, and applied it to the spiral problem. He used 29 perceptrons but had difficulties capturing the structure of the spirals due to the piecewise linearity of his decision boundaries.

The two-spiral problem was chosen for this study because it is a hard problem for backpropagation networks due to high non-linearity, it is a noise-free problem, and the generalization performance of different predictors can be easily visualized on the two-dimensional plane.

In Section 5, we demonstrate our method on another well-known machine-learning problem, the prediction of coronary artery disease based on the Cleveland heart data, which reside in the University of California at Irvine (UCI) machine-learning repository (Murphy & Aha, 1992).

4. Results on the Spiral Data

4.1. *Feedforward Network Architecture*

Downloaded by [ ] at 03:16 22 September 2011

We used Ripley's (1996) S-Plus NNET package, which implements backpropagation. The minimization criterion is MSE with weight-decay regularization of the form

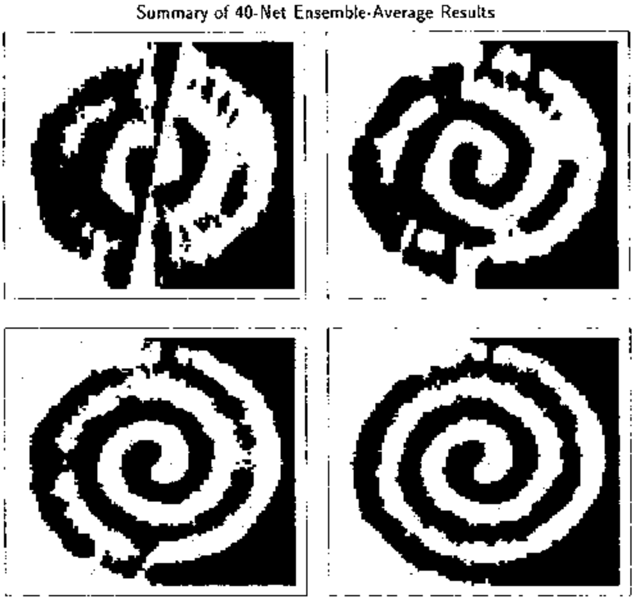
$$E = \sum_{p} |t_p - y_p|^2 + \lambda \sum_{i,j} w_{i,j}^2$$

where t_p is the target and y_p the output for the pth example pattern. $w_{i,j}$ are the weights and λ is a parameter that controls the amount of weight decay regularization.

The network architecture was 2–30–1 (two inputs, 30 hidden units and one output). The first and last layers were fully connected to the hidden layer giving a total of 121 weights. The transfer function of the hidden and output units was the logistic sigmoidal function. The initial weights were random from $U(-0.7, 0.7)$. It should be noted here that although we are training 5–40 networks, the effective number of parameters is not more (and probably even less) than the number of parameters for a single network. This is because we do not have the flexibility to estimate an optimal combination of predictors, but rather take the simple average of them.

Baseline results were obtained by training 40 networks without any regularization. We derived then an average predictor whose output is the mean of all the 40 nets' outputs (Figure 2 (top left)). The predictor had no smoothness constraints and therefore found relatively linear boundaries (this can also be seen in Figure 3 (top left), where a five-net ensemble average is taken).

4.1.1. Effect of training with noise on a flexible predictor. We trained 30 hidden-unit networks using the bootstrap method (as described in Section 3), with noise SD ranging from $\varepsilon = 0$ to $\varepsilon = 0.8$, and $K = 10N$. Figure 3 demonstrates the effect of noise on the predictor. Each image is a threshold output of a five-net ensemble average predictor. Noise level goes from $\varepsilon = 0$ in the upper left image through $\varepsilon = 0.8$ in the lower right. The classification results are drawn on a uniform grid of 100×100 points (namely, a much larger test set) so as to get a clear view of the



Downloaded by [ ] at 03:16 22 September 2011

Figure 2. Summary of 40-net ensemble results. *Top left:* No constraints (no weight decay or noise). *Top right:* Optimal weight decay ($\lambda = 3e - 4$) and no noise. *Bottom left:* Optimal noise (Gaussian SD $= 0.35$) and zero weight decay. *Bottom right:* Optimal noise and optimal weight decay. The classification threshold in this figure and the following ones is 0.5.

classification boundaries defined by the classifier. It can be seen that for small noise levels ε, the ensemble average predictor is unable to find any smooth structure in the data and merely over-fits to the training data. For moderate levels of noise, a better structure can be found, and for large levels of the noise, the data are so corrupted that again no structure can be found. The optimal noise SD was around $\varepsilon = 0.35$.

4.1.2. Effect of weight-decay regularization. Weight-decay regularization involves finding an optimal parameter λ that controls the amount of weight decay versus the bias of the net. We trained networks with different λ's and found that optimal values were around $\lambda = 3e - 4$. When comparing the effect of averaging alone with the effect of regularization via weight decay with no averaging, it turns out that the bootstrap method (averaged over different initial network weights) has better generalization properties than the weight-decay method. The weight-decay regularization does not generalize well on the outer points, where the training data are more sparse.

4.1.3. Applying bootstrap to networks with weight decay. Our best results were obtained when applying the BEN method to networks with optimal weight-decay regularization. Figure 4 demonstrates the effect of bootstrap with noise on the performance of a five-net ensemble trained with optimal weight decay. The effect of ensemble averaging over networks that were trained with different random initial conditions only is demonstrated in the top left image which represents no

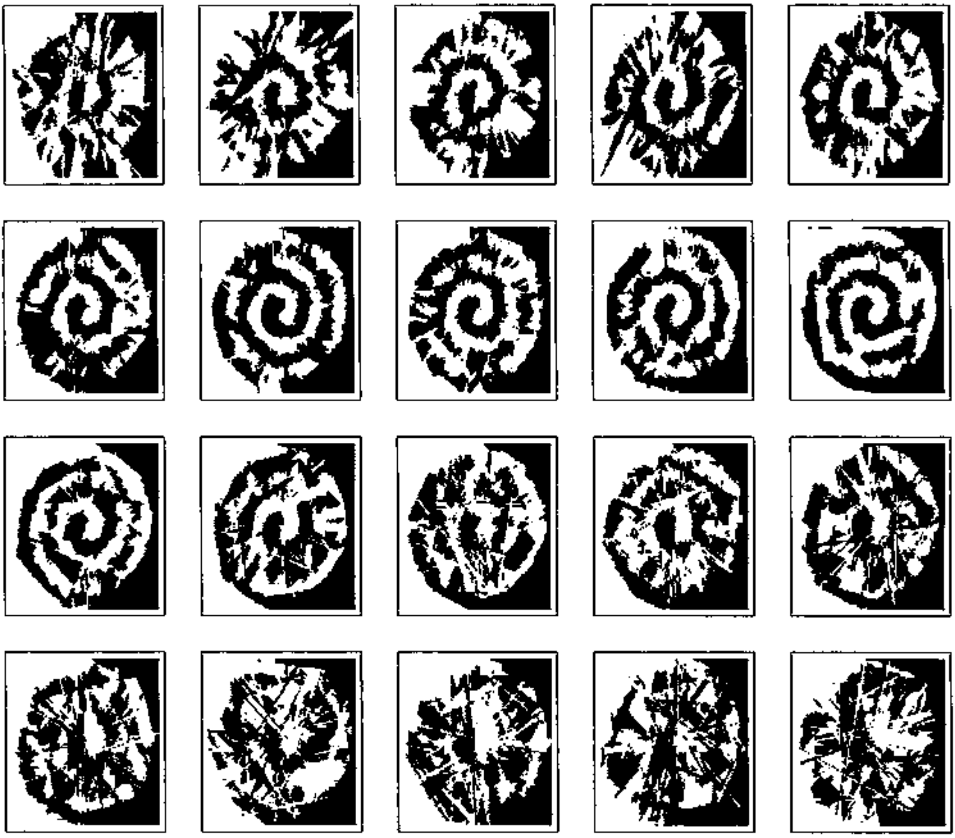


Figure 3. Effect of training with different levels of Gaussian noise. Ensembles of five networks with no weight decay and a varying degree of noise (top left is zero noise, bottom right is noise with SD = 0.8).

Downloaded by [ ] at 03:16 22 September 2011

noise during training. Optimal noise values are similar to those obtained when training with no weight decay, and are surprisingly high (see Figure 1 (right) for the corruption of noise to the data). Although the results look better than those with no weight decay, in the sense that the boundaries look smoother, they can still be improved by averaging on a larger ensemble of networks. This is demonstrated in the next section (Figure 2).

The effect of averaging is summarized in Figure 2. It can be seen that the 40-net ensemble averaging results, with no weight decay and no noise are better than the corresponding ones when an ensemble of five nets is used (Figure 3). Similarly, the results for an ensemble of 40 networks trained with optimal weight decay with no noise are better than the corresponding five-net ensemble (Figure 4 (top left)). Finally, the combination of weight decay, noise and 40-net ensemble clearly gives the best results (Figure 2 (bottom right)). Thus, while earlier work suggested that a single-layer feedforward network is not capable of capturing the structure in the spiral data, it is evident that a network ensemble with strong control over its capacity (via weight decay) which is trained with heavy noise can discover the highly non-linear structure of the problem.

Downloaded by [ ] at 03:16 22 September 2011

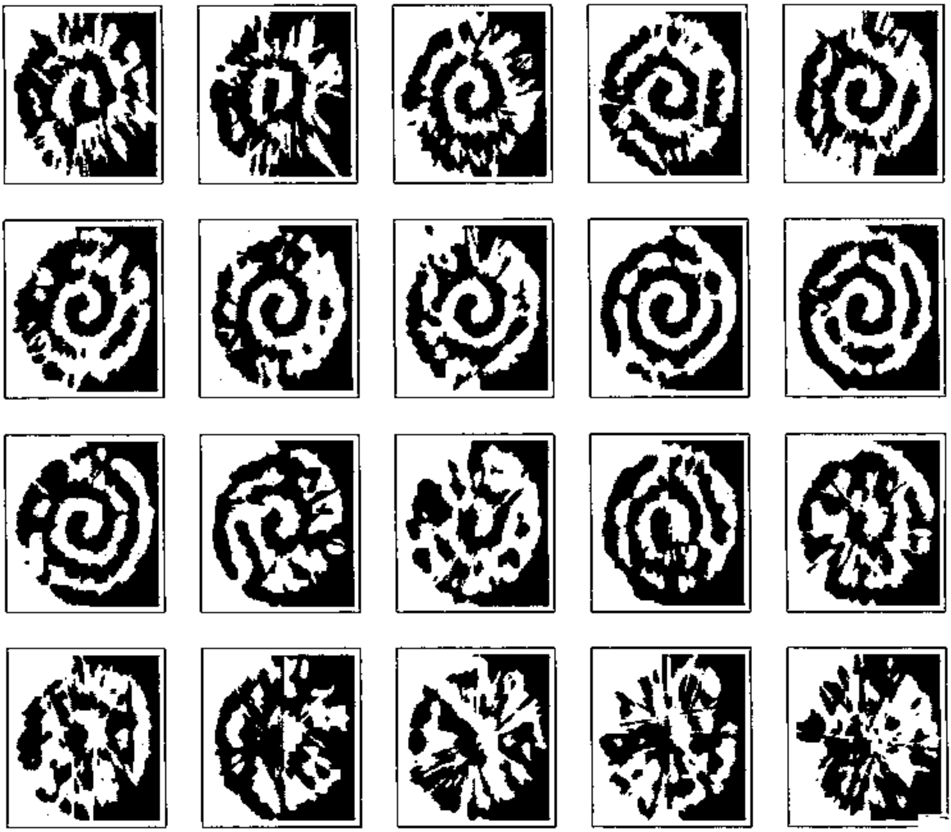


Figure 4. Effect of training with different noise levels on five-net ensemble networks with weight decay. Noise levels are as before, 0–0.8 from top left to bottom right.

4.2. Generalized Additive Models

In this section, we take a different approach. Instead of analyzing a method that has a hard time with the spiral data, we study a model that is very natural for it. We apply bootstrapping to a generalized additive model (GAM) (Hastie & Tibshirani, 1986, 1990) with a polynomial fit of degree 1 on the same data. We had to optimize the degree of the polynomial and the span degree, which determines the smoothness and the degree of locality of the estimation.[3] Due to these efficient controls, this flexible model is much more appropriate for the spiral data. Furthermore, this algorithm provides a unique model, i.e. for each set of parameters, there is no variability in the produced models as opposed to the variability generated by the random initial weights of a feedforward network. All of this suggests that there should be no reason to bootstrap with noise, since the smoothness and locality already can control the smoothness of the boundary surface, and there seems no reason to corrupt the data with unfamiliar noise. Moreover, there is no need to average over several models since there is no variability due to different local minima of the resulting model.

It is thus surprising that even in this extreme case, bootstrapping with noise improved the generalization results. Figure 5 depicts the results for various degrees of noise added during training. It is clear that the bootstrap improves results, and, furthermore, small values of the noise sharpen the result.

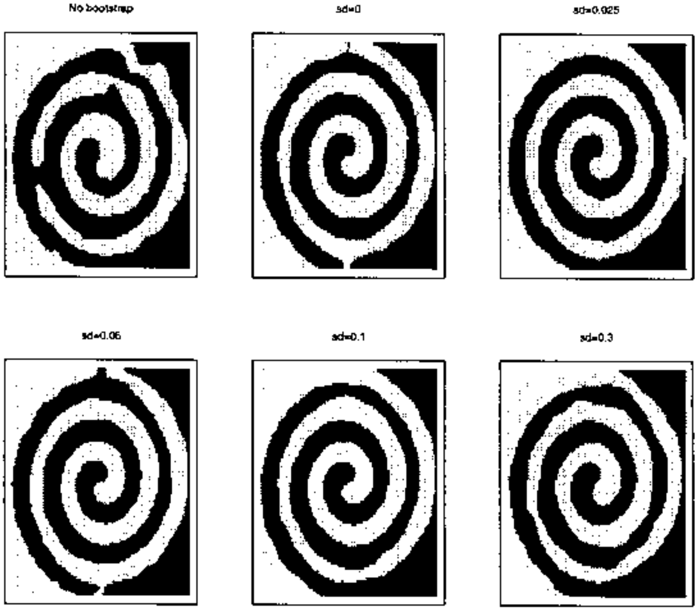


Figure 5. Model estimation using GAM with bootstrap. Ten GAM predictors are averaged using bootstrap samples with varying degree of noise. There is no noise (and thus no averaging) at the top left result.

5. Cleveland Heart Data

In this section, we analyze the Cleveland heart data (Detrano *et al.*, 1989), donated by Dr Robert Detrano[4] to the UCI machine-learning repository (Murphy & Aha, 1992). This data concerns diagnosis of coronary artery disease and has been used in the past by statisticians and by the machine-learning community (Brazdil & Henery, 1994; Detrano *et al.*, 1989; Gennari *et al.*, 1988; Stensmo, 1995). Further data and pre-processing details are given in Appendix B. The pre-processing, which included removal of missing values, sphering the data and creating dummy variables to replace categorial variables, resulted in a dramatic improvement over past results. Moreover, it revealed that in the new data representation, the structure is very linear since logistic regression was able to obtain a nine-fold cross-validation error of about 15.2%. A similar error was obtained by using extensive pre-processing and temporal-difference reinforcement learning (Stensmo, 1995). Both results are consistent with our feedfoward architecture results with no noise injection and are (as far as we know) the current best results on this data.[5]

It is thus a very challenging problem to NNs as deviation from linear structure is very small,[6] and highly non-linear estimators such as CART, radial-basis functions and KNN did not do so well on this data (Brazdil & Henery, 1994). The problem is complementary to the spiral problem that was considered before; there, we attempted to improve performance on a highly non-linear data which required a large capacity network, while here we try to improve performance on a relatively linear problem using a small capacity network. In both cases, we show that noise cannot be replaced by network size or weight-decay regularization and is essential for good performance.

Downloaded by [ ] at 03:16 22 September 2011

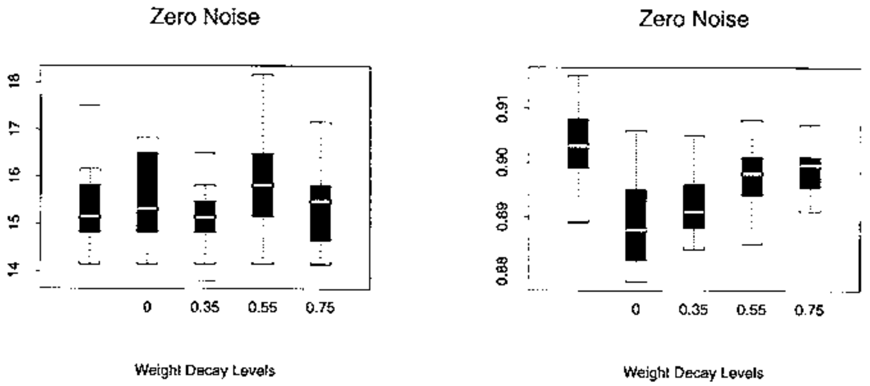


Figure 6. Results from logistic regression and from feedforward networks with three hidden units and varying degrees of weight decay. *Left:* Per cent classification error. *Right:* ROC values. All results were obtained with nine-fold cross-validation on the Cleveland heart data. In both graphs, the first boxplot from the left represents the generalized linear model results.

Downloaded by [ ] at 03:16 22 September 2011

Figure 6 summarizes model comparison of results between logistic regression and nine-fold cross-validation,[7] with three hidden-unit networks based on Ripley's NNET package described in Section 4.1. Training was stopped after 400 epochs or earlier, based on Ripley's conditions. The network results were obtained by training five networks on each of the nine-fold cross-validation sets and averaging their results. Thus, each classification error is generated out of 45 networks. In each of the following figures, the statistics were obtained from 12–20 similar runs differing in random initial conditions and choice of cross-validation sets from the data. The cross-validation code is based on the public domain version of Tibshirani in Statlib.[8] The results are summarized by boxplots[9] (Hoaglin *et al.*, 1983). Each boxplot is based on 500–900 single network runs. As the ratio between the two classes is different than one, classification results are not a very robust measure for model comparison, since they are based on a single classification threshold. For example, if one class represents only 10% of the data, then setting up the threshold to 1 will result in a trivial classifier that will produce zero regardless of the input and will have only 10% error. The receiver-operating characteristic (ROC) (Goodenough *et al.*, 1974; Hanley & McNeil, 1982) is frequently used in such model comparisons, especially in clinical data (Henderson, 1993). This measure has been used by the contributor of the data (Detrano, 1989) and in assessing neural network performance on other heart disease data (Lippmann *et al.*, 1995).

Figure 6 implies that the performance of NNs (without noise injection) as measured by error rate and ROC values are slightly worse (not statistically significant) compared with logistic regression, and cannot be improved by weight-decay regularization alone.

Figure 7 shows the effect of noise injection for various levels of weight decay for an over-capacity architecture of nine hidden units. Noise levels in all the following graphs represent the SD of the zero-mean Gaussian noise. Although noise injection produces significant improvement, the absolute values are sub-optimal since the architecture is too large. Note, however, that the ROC values for the 0.75 weight decay net are the highest compared with logistic regression

Downloaded by [ ] at 03:16 22 September 2011

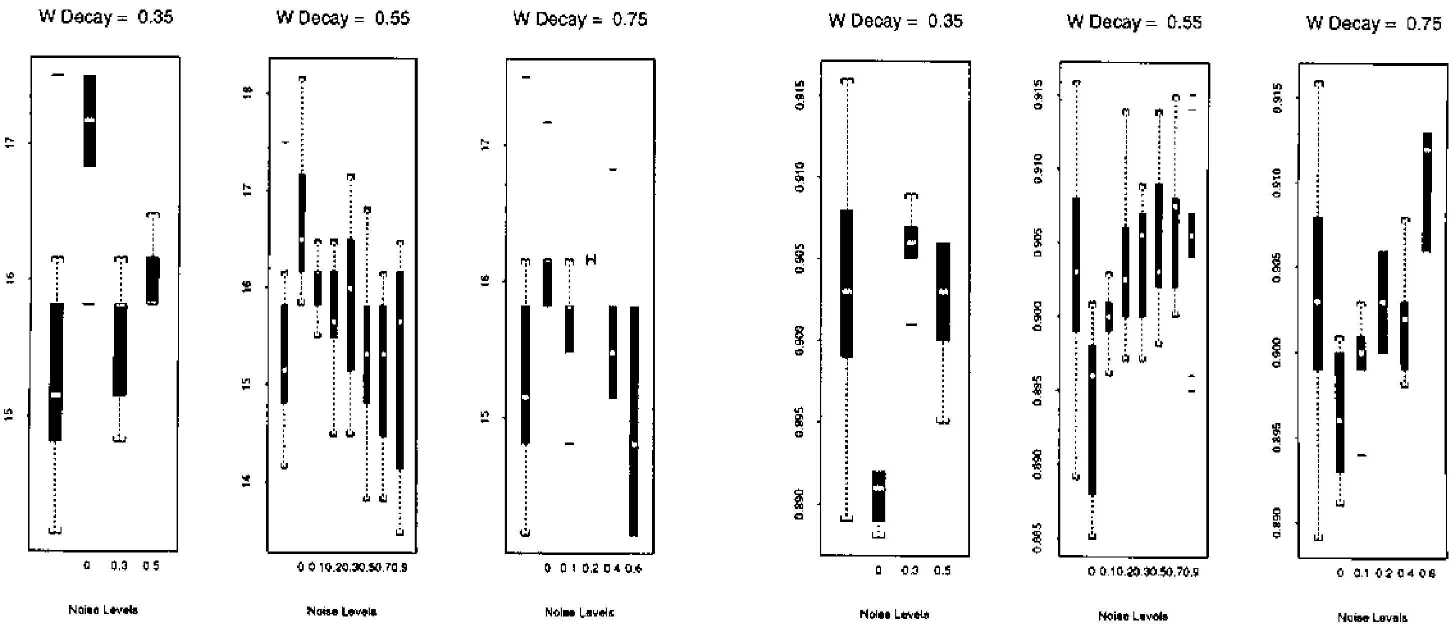


Figure 7. Integrated bias, variance and total error for the hepatoma data set.

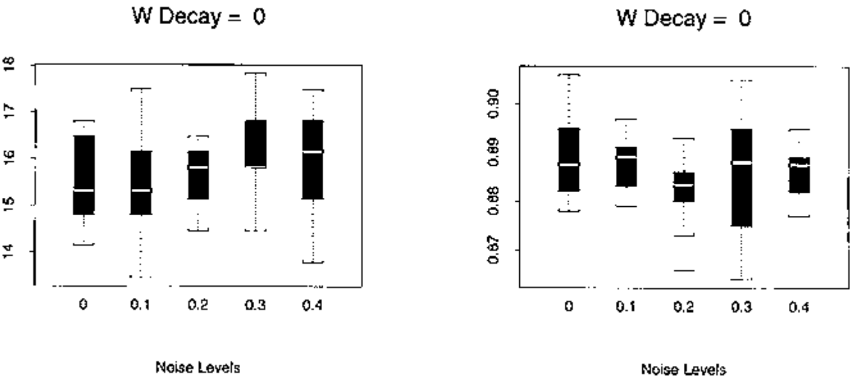


Figure 8. The effect of noise injection is diminished when no weight decay is used (compare with Figure 9). An optimal architecture of three hidden units cannot produce good results without weight decay. *Left:* Classification error. *Right:* ROC values.

Downloaded by [ ] at 03:16 22 September 2011

(GLM − ROC = 0.903 ± 0.001, NNET − ROC = 0.91 ± 0.002; $t = 1.766$, degrees of freedom (df) = 21, $P < 0.045$; $Z = 1.691$, $P < 0.045$) or with the optimal three hidden-unit network. We have been using both the t-statistic (Hogg & Craig, 1970) and the Z-statistic of the Wilcoxon test (Lehmann, 1975) which uses a non-parametric rank to test the difference in the medians, as it is more robust to outliers. The ROC results suggest that the classification error of this model could be improved, possibly by averaging over a larger number of networks. To see the performance of noise injection alone, we present results of noise injection into zero weight-decay, optimal architecture (Figure 8) and show that even under a low-capacity architecture, weight decay is essential to stabilize the system.

Optimal results are presented in Figure 9. With optimal weight decay and architecture, addition of noise achieves results which are better than any other network, and better than logistic regression. Mean error of logistic regression was 15.27 ± 0.18, mean error for zero-noise net was 15.07 ± 0.13 and mean error for noise with SD = 0.3 was 14.56 ± 0.22. The difference between the optimal neural network and logistic regression is statistically significant ($t = 2.196$, df = 26,

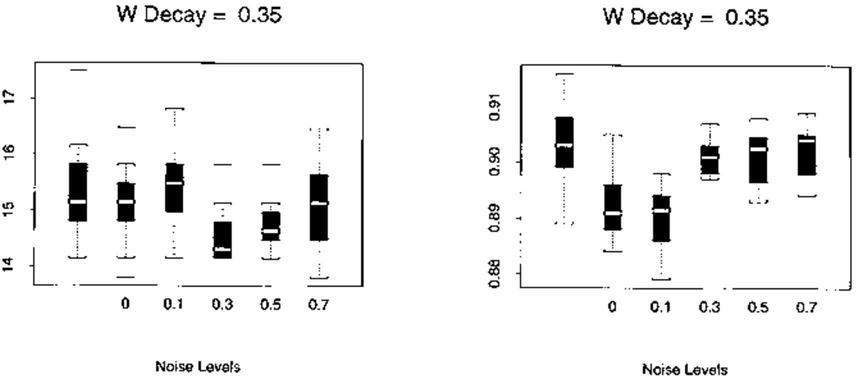


Figure 9. Results for the optimal architecture network. *Left:* Classification error. *Right:* ROC values. Noise injection is helpful and overall performance is optimal.

$P < 0.018$; $Z = 2.14$, $P < 0.016$) and the difference to zero noise is significant as well ($t = 2.045$, df $= 27$, $P < 0.025$; $Z = 2.029$, $P < 0.021$). To our knowledge, these are the best results on the Cleveland heart data.

6. Discussion

The motivation to our approach comes from a key observation regarding the bias/variance decomposition of prediction error, namely the fact that ensemble averaging does not affect the bias portion of the error, but reduces the variance, when the estimators on which averaging is done are independent. The level of noise affects the independency between the training sets, and thus the relative improvement of ensemble averaging. However, the level of noise also affects the quality of each predictor separately, increasing its variance by increasing the variability in the data. Thus, there should be an optimal level of the noise (it may not correspond to the true noise), which leads to optimal ensemble performance. This performance can be further improved if the variance of individual networks can be tempered, e.g. with weight decay.

We have demonstrated the effect of noise injection on prediction in three different cases. (i) Highly non-linear (spiral) data, using a non-appropriate model (as the data are almost radially symmetric and the neural net is not). This required the use of an ensemble of high capacity single predictors and thus made the regularization task challenging. It was shown that the excess variance of high capacity models could only be effectively trimmed by a combination of all three components: weight decay, noise injection and ensemble averaging. (ii) Highly non-linear (spiral) data with essentially the perfect model for it (GAM with locally linear units). Even in this case, where regularization provides the perfect bias to the model, performance could be improved by the combination. (iii) A highly linear problem, where practically any network has excess capacity. This case is a representative of a family of clinical data sets, in which (linear) variable selection was applied to highly dimensional data and resulted in a highly linear low-dimensional data structure. It was thus challenging to be able to show that the BEN algorithm is useful in this case, and can lead to improved classification results. Performance was also evaluated based on the ROC measure, as it is a standard model comparison tool for clinical data analysis.

The theoretical analysis suggests that it is best to start with a very flexible function approximation technique (e.g. a feedforward network with a large number of hidden units) and then control its capacity and smoothness using noise and averaging. Our conclusions are not restricted to artificial neural network estimation. We show that similar conclusions can be obtained when using a highly flexible GAM (Hastie & Tibshirani, 1986).

Downloaded by [ ] at 03:16 22 September 2011

Acknowledgements

Stimulating discussions with Leo Breiman, Brian Ripley and Chris Bishop are gratefully acknowledged.

Notes

[1] An example of an identifiable model is (logistic) regression.

[2] Where $\mathrm{Var}(f)$ is defined by $E_{\mathcal{D}}[(f_{\mathcal{D}}(x) - E_{\mathcal{D}}[f_{\mathcal{D}}(x)])^2]$.

3. In this case, the model amounts to a sum of locally linear functions around each of the training samples.
4. VA Medical Center, Long Beach and Cleveland Clinic Foundation.
5. Recent best result of 23.1% on non-normalized data was obtained by a company that provides classification with its own proprietary software (UDM, 1996).
6. This is a classical problem in clinical data in which variable selection was done by a linear method and therefore the data contains mostly variables with linear structure.
7. This is a standard use; see, for example, results under the STATLOG ESPRIT project (Brazdil & Henery, 1994).
8. http://www.stat.cmu.edu.
9. To read the boxplot: the white line in the middle of the box represents the median of the distribution; the grey box represents the inter-quartile range such that the bottom of the box is the first quartile and the top is the third quartile; the dashed line and its terminating line represent plus and minus 1.5 inter-quartile distance from the median; points lying outside this range are considered outliers, each such point is represented by a whisker.
10. Can be obtained from Murphy and Aha (1992).

References

Baum, E. & Lang, K. (1991) Constructing hidden units using examples and queries. In R. P. Lippmann, J. E. Moody & D. S. Touretzky (Eds), *Advances in Neural Information Processing Systems*, Vol. 3. San Mateo, CA: Morgan Kaufmann, pp. 904–910.

Bishop, C.M. (1995) Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, **7,** 108–116.

Brazdil, P. & Henery, R. (1994) Analysis of results. In D. Michie, D. J. Spiegelhalter & C. C. Taylor (Eds), *Machine Learning, Neural and Statistical Classification*. New York: Ellis Horwood, pp. 175–212.

Breiman, L. (1994) *Bagging Predictors.* Technical Report TR-421, Department of Statistics, University of California, Berkeley, CA.

Deffuant, G. (1995) An algorithm for building regularized, piecewise linear discrimination surfaces: the perceptron membrane. *Neural Computation*, **7,** 480–489.

Detrano, R. (1989) Accuracy curves: An alternative graphical representation of probability data. *Journal of Clinical Epidemiology*, **42,** 983–986.

Detrano, R., Janosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J., Sandhu, S., Guppy, K., Lee, S. & Froelicher, V. (1989) International application of a new probability algorithm for the diagnosis of coronary artery disease. *American Journal of Cardiology*, **64,** 304–310.

Efron, B. & Tibshirani, R. (1993) *An Introduction to the Bootstrap.* New York: Chapman and Hall.

Fahlman, S.E. (1989) Fast-learning variations on back-propagation: An empirical study. In D. Touretzky, G. Hinton & T. Sejnowski (Eds), *Proceedings of the 1988 Connectionist Models Summer School*. San Mateo, CA: Morgan Kaufmann, pp. 38–51.

Fahlman, S.E. & Lebiere, C. (1990) *The Cascade-Correlation Learning Architecture.* Cmu-cs-90-100, Carnegie Mellon University, Pittsburgh, PA.

Geman, S., Bienenstock, E. & Doursat, R. (1992) Neural networks and the bias-variance dilemma. *Neural Computation*, **4,** 1–58.

Gennari, J.H., Langley, P. & Fisher, D. (1988) Models of incremental concept formation. *Artificial Intelligence*, **40,** 11–61.

Goodenough, D.J., Rossmann, K. & Lusted, L.B. (1974) Radiographic applications of receiver operating characteristic (ROC) curves. *Radiology*, **110,** 89–95.

Hanley, J.A. & McNeil, B.J. (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, **143,** 29–36.

Hansen, L.K. & Salamon, P. (1990) Neural networks ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12,** 993–1001.

Hastie, T. & Tibshirani, R. (1986) Generalized additive models. *Statistical Science*, **1,** 297–318.

Hastie, T. & Tibshirani, R. (1990) *Generalized Additive Models.* London: Chapman and Hall.

Henderson, A.R. (1993) Assessing test accuracy and its clinical consequences: a primer for receiver operating characteristic curve analysis. *Annals of Clinical Biochemistry*, **30,** 521–539.

Hoaglin, D.C., Mosteller, F. & Tukey, J.W. (1983) *Understanding Robust and Exploratory Data Analysis.* New York: Wiley.

Hogg, R.V. & Craig, A.T. (1970) *Introduction to Mathematical Statistics* (3rd edn). Toronto, Canada: Macmillan.

Downloaded by [ ] at 03:16 22 September 2011

Krogh, A. & Hertz, J.A. (1992) A simple weight decay can improve generalization. In J. E. Moody, S. J. Hanson & R. P. Lippmann (Eds), *Advances in Neural Information Processing Systems*, Vol. 4. San Mateo, CA: Morgan Kaufmann, pp. 950–957.

Lang, K.J. & Witbrock, M.J. (1988) Learning to tell two spirals apart. In D. S. Touretzky, J. L. Ellman, T. J. Sejnowski & G. E. Hinton (Eds), *Proceedings of the 1988 Connectionists Models*, pp. 52–59.

Lehmann, E.L. (1975) *Nonparametrics: Statistical Methods Based on Ranks.* San Francisco, CA: Holden and Day.

Lippmann, R.P., Kukolich, L. & Shahian, D. (1995) Predicting the risk of complications in coronary artery bypass operations using neural networks. In G. Tesauro, D. Touretzky & T. Leen (Eds), *Advances in Neural Information Processing Systems,* Vol. 7. Cambridge, MA: MIT Press, pp. 1055–1062.

Murphy, P.M. & Aha, D.W. (1992) UCI Repository of machine learning databases. Department of Information and Computer Science, University of California at Irvine.

Perrone, M.P. (1993) *Improving Regression Estimation: Averaging Methods for Variance Reduction with Extensions to General Convex Measure Optimization.* PhD thesis, Brown University, Institute for Brain and Neural Systems, Providence, RI.

Ripley, B.D. (1996) *Pattern Recognition and Neural Networks.* Oxford Press.

Sietsma, J. & Dow, R.J.F. (1991) Creating artificial neural networks that generalize. *Neural Networks*, **4**, 67–79.

Stensmo, M. (1995) *Adaptive Automated Diagnosis.* PhD thesis, Royal Institute of Technology, Stockholm, Sweden.

Ultragem Data Mining (UDM) (1996) *Esprit Statlog Benchmarks.* Technical Report, Boulder Creek, CA.

Wolpert, D.H. (1992) Stacked generalization. *Neural Networks*, **5**, 241–259.

Downloaded by [ ] at 03:16 22 September 2011

Appendix A: The Spiral Data

The two-dimensional spiral data[10] (Lang & Witbrock, 1988) are given by a vector (x_i, y_i) defined by:

$$x_i = r_i \, cos(\alpha_i + k\pi/2), \qquad y_i = r_i \, sin(\alpha_i + k\pi/2) \tag{A1}$$

where

$$\alpha_i = \pi i/16, \qquad r_{\mathrm{i}} = 0.5 + i/16, \qquad i = 0, \ldots, 97 \tag{A2}$$

and $k = 1$ for one class and 3 for the other class.

Appendix B: Details and Pre-processing of the Cleveland Heart Data

The data in the UCI repository contain 13 variables out of about 70 that were in the original study. The task is to predict the existence of a coronary artery disease (CAD) based on the measurements. Data for 303 patients were obtained; 44% of the patients were diagnosed with CAD. The variable attributes are:

(1) Age
(2) Sex
(3) Chest pain type (4 values—converted to 3 binary variables)
(4) Resting blood pressure
(5) Serum cholesterol in mg dl^{-1}
(6) Fasting blood sugar > 120 mg dl^{-1}
(7) Resting electrocardiographic results (values 0, 1, 2)
(8) Maximum heart rate achieved
(9) Exercise-induced angina
(10) Oldpeak = ST depression induced by exercise relative to rest

(11) The slope of the peak exercise ST segment (converted to 2 binary variables)
(12) Number of major vessels (0–3) coloured by flouroscopy (converted to 3 binary variables)
(13) Thal: 3 = normal; 6 = fixed defect; 7 = reversible defect (converted to 2 binary variables)

We have added dummy variables to replace the categorial and ordinal variables for variables 3, 11, 12 and 13 and therefore worked with 19 independent variables. The continuous variables 1, 4, 5, 8 and 10 were sphered (standardized) by setting the mean of each of the variables to zero with unit variance. This step was necessary as the data contain variables that are on different scales, such as age and blood pressure. The original data contain 76 attributes and have many missing values. The data used in most of the benchmarks have only 13 attributes and a few missing values which we simply replaced by their unconditional expectations. The addition of dummy variables and data sphering had a dramatic effect on the classification results.

Downloaded by [ ] at 03:16 22 September 2011