*To appear in the proceedings of*
*Interface 97. 29th Symposium on the Interface: Computing Science and Statistics*

# Can Statistical Theory Help Us Use Neural Networks Better?

B.D. Ripley
Department of Statistics
University of Oxford
Oxford OX1 3TG, UK
ripley@stats.ox.ac.uk, http://www.stats.ox.ac.uk/∼ripley/

## Abstract

If we view neural nets as a class of statistical models with high-dimensional parameters, we can consider how to apply the ideas of statistical theory, in particular ideas for model choice and the concepts of predictive Bayesian inference. It turns out that these ideas give considerable insight, and enable us to find more powerful solutions with reduced computational load. This is illustrated by two case studies.

## 1 Types of Neural Network

One type of neural network predominates, known variously as a *multi-layer perceptron*, a *backpropagation network* or a *feed-forward network* (FFNN). The other methods than seem to be used at all seriously are *Radial basis function networks* (RBFs) and Kohonen's *Self Organizing Maps*. RBFs are used in very similar ways to FFNNs, whereas SOMs are a variant of cluster analysis or multidimensional scaling. Ripley (1996) provides a wide-ranging overview of the area; more of the background philosophy may be found in Ripley (1993) and standard neural network texts. [Of the latter, Bishop (1995) is the most recommendable but is narrow in scope; Haykin (1994) is encyclopaedic and shallow, Hertz *et al.* (1991) and Kung (1993) are sounder and mathematically more detailed, but becoming dated. These texts, however, are famously free of realistic examples, and indeed on details of how to *use* neural networks.]

There are some published comparative studies between neural networks and other methods, including those in Weiss & Kulikowski (1991) and Michie *et al.* (1994), although many of these make poor use of statistical or neural network methods (or both).

### What is a FFNN?

FFNNs are usually represented by a picture such as figure 1. In mathematics:

$$y_k = \phi_o\Big(\alpha_k + \sum_j w_{jk}\,\phi_h\Big(\alpha_j + \sum_i w_{ij}\,x_i\Big)\Big)$$

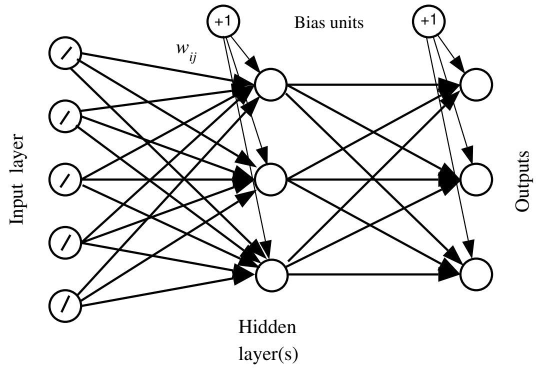



Figure 1: A generic feed-forward neural network.

where the parameters $w_{ij}$ and $\alpha_i$ are called *weights* and

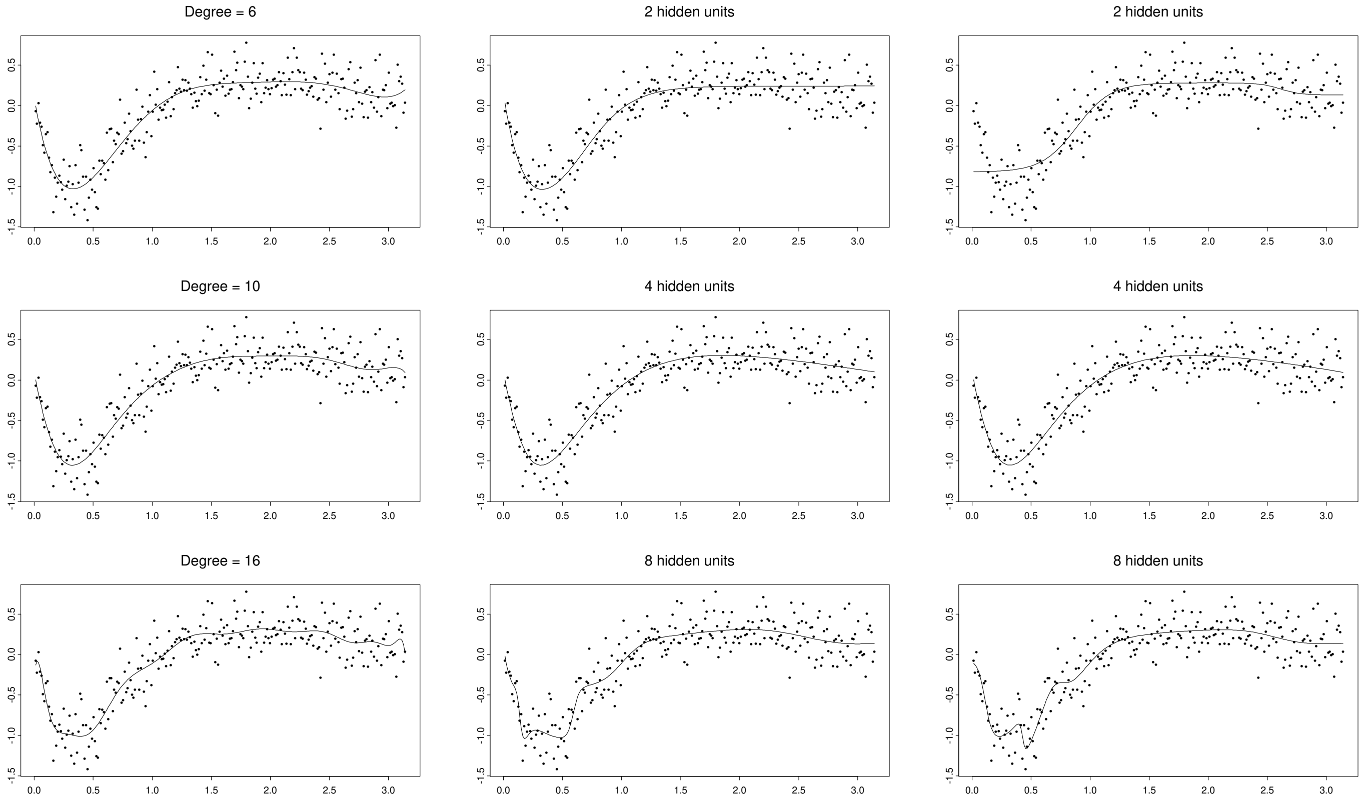
- $\phi_h$ is usually the logistic $\phi_h(x) = e^x/(1+e^x)$
- $\phi_o$ is linear, logistic or threshold
- the 'biases' $\alpha_i$ can be replaced by weights to $+1$
- the number of *hidden units* can vary
- sometimes weights are set to zero or equal.

An obvious extension is to include 'skip-layer' connections from input to output to allow a linear part of the fitted function:

$$y_k = \phi_o\Big(\alpha_k + \sum_i w_{ik}x_i + \sum_j w_{jk}\,\phi_j\Big(\alpha_j + \sum_i w_{ij}\,x_i\Big)\Big)$$

(This is equivalent to scaling down input weights and up output weights for one hidden unit, but can be set initially from a linear fit.)

Statisticians may like to think as this as a regression of logistic regressions plus the original variables.




(a) Polynomial fits

(b) FFNN fits

(c) More FFNN fits

Figure 2: Curve-fitting by polynomials and by feed-forward neural networks.

### Radial Basis Functions

RBFs are approximations of the form

$$f(\boldsymbol{x};\theta) = \alpha + \sum_j \beta_j G(\|\boldsymbol{x} - \boldsymbol{x}_j\|)$$

for (pre-specified) centres $\boldsymbol{x}_j$. They come from the field of approximation theory (Powell, 1987).

Examples of $G$ proposed include

- the Gaussian $G(r) = \exp -r^2/2$
- the multiquadric $G(r) = \sqrt{(c^2 + r^2)}$ (Hardy, 1971, 1990)
- the thin-plate-spline function $G(r) = r^2 \log r$.

### Universal approximation

Both FFNNs and RBFs provide functions $g\colon \mathbb{R}^p \to \mathbb{R}^k$, of 'complexity' $N$ and some parameters $\theta$. Both are *universal approximators* in the sense that the family of functions is dense in the topologies of uniform convergence on compacta and in $L_q(\mu)$ for any probability measure $\mu$ on $\mathbb{R}^p$, and any $q$. There are even results on the rate of convergence in $L_2(\mu)$, $O(1/\sqrt{N})$ (Barron, 1993,4; Girosi & Anzelloti, 1993), but they apply to quite restricted classes of target functions.

In other words, they can in principle model anything (noise as well as signal).

## 2 'Training' a FFNN

The process of selecting the parameters of a FFNN is usually known as 'training'. This is normally done by least-squares or maximum likelihood, although unconventional optimization algorithms are used.

What's different about neural networks, compared to curve-fitting as done in statistics? The main differences are:

(a) Multiple local minima are common (especially for large models).

(b) Over-fitting is controlled by non-likelihood fitting.

(c) Sequence of possible architectures is not necessarily nested.

The consequence is that everyone fitting polynomials will get the same answer, but almost no two people fitting neural networks will. (*Exception:* networks without hidden layers are linear and usually have a unique solution.) This is illustrated in figure 2, which shows that slight differences in the starting values for the numerical optimization can give quite different fits.

## Demonstration example

Figure 3 shows a simulation of an example taken from Wahba & Wold (1975) where it illustrates smoothing splines. This has 100 points whereas figure 2 had 250 points. Throughout we assume that the true noise variance, $(0.2)^2$, is known.

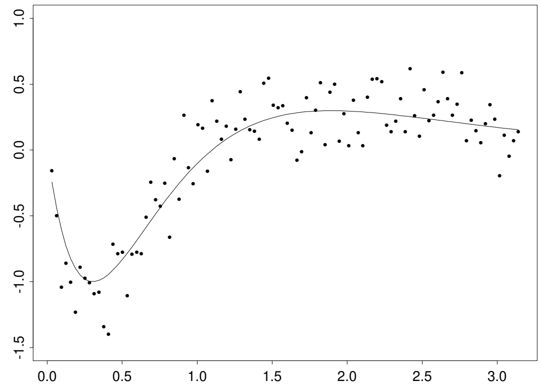


Figure 3: 100 data points from a curve-fitting problem, with the true curve.

Model selection amounts to choosing the number of hidden units. Figure 4 shows neural-net fitting with 8 hidden units treated as a non-linear regression problem, with the standard confidence limits produced by linearization (Bates & Watts, 1988). Figure 5 shows that there are quite a number of different local minima; all these solutions fit about equally well.

We clearly have over-fitting; there are many ways to avoid this. Perhaps the best is to use *regularization* as used for smoothing splines. Somewhat easier for neural networks to use *weight decay*:

$$\sum_{i=1}^{n}[y_i - f(x_i;\mathbf{w})]^2 + \lambda \sum_{ij} w_{ij}^2$$

with the inputs and outputs suitably scaled. In theory we only need weight decay on the input-to-hidden weights (parameters), as these control the slope of the logistic terms and hence

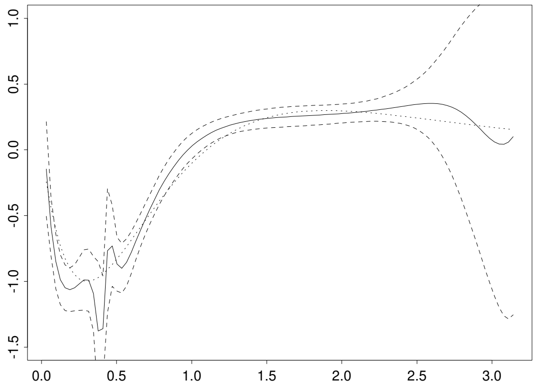


Figure 4: Curve fitted by 1-8-1 neural network, with $\pm 2$ standard error bands (computed as a non-linear regression problem) and true curve (dotted).

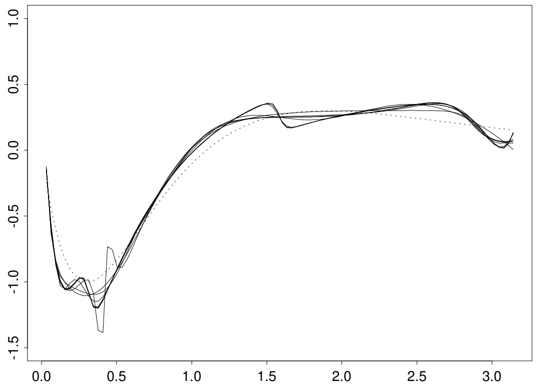


Figure 5: 1-8-1 neural networks with 25 different sets of starting values for the optimization.

the 'roughness' of the fitted functions. However, weight decay on the terms leading to the output unit(s) is analogous to ridge regression and is also beneficial.

### *Bias/variance Compromises*

We now have two ways to control the fit, the number of hidden units and $\lambda$, the degree of regularization. The true curve is not exactly representable by our model, so we have some bias (figure 7). Choosing fewer hidden units leads to more bias, as does adding a regularizer. But it also reduces their variability, so reduces the mean-square error. Note that the error bands in figure 4 are smaller than those in figure 6; this is misleading as the conventional local linearization used for figure 4 is not sufficiently accurate.

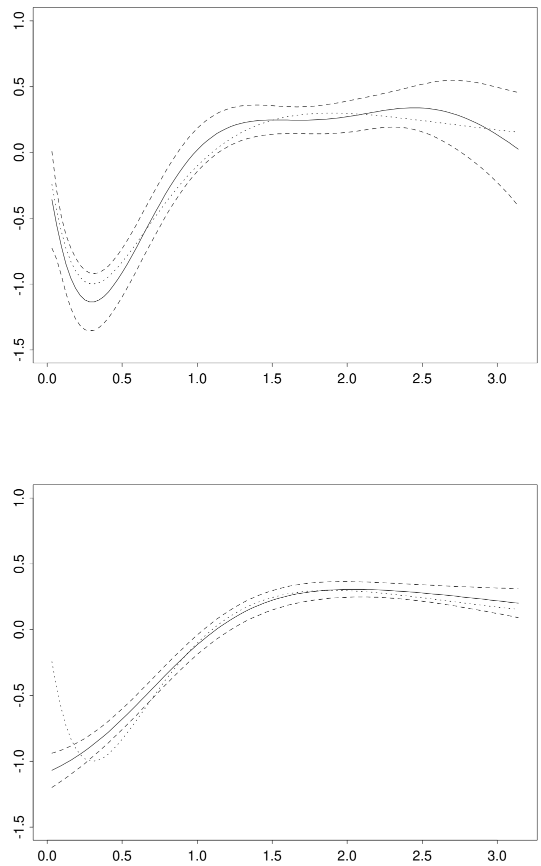



Figure 6: The effect of adding weight decay. Curves fitted by 1-8-1 neural network, with $\pm 2$ standard error bands and true curve $\lambda = 10^{-3}$ (top) and $\lambda = 0.1$ (bottom).

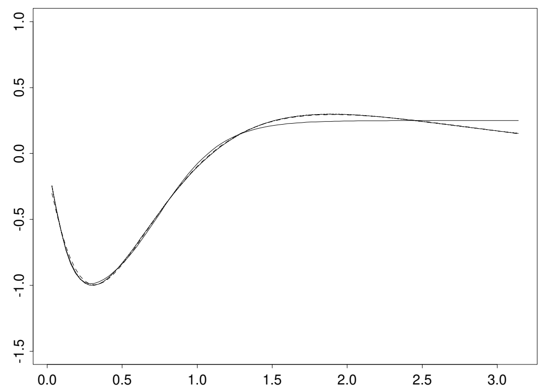



Figure 7: Fits to the *true curve* with 2 and 3 hidden units, and 8 units with $\lambda = 10^{-3}$.

## 3 Flexible Families for Discrimination

Suppose we are given a rather flexible family $f(x;\theta)$ of functions, how do we use it to do discrimination? How do we use linear functions? Ripley (1994a. 1996) describes the following ideas.

(a) We model the log posterior probabilities

$$\log P(C = c \mid X = x) = g_c(x;\theta)$$

and fit by maximum (conditional) likelihood. Allocate future examples to class with highest $g_c(x;\widehat{\theta})$.
[known as *logistic discrimination*]

(b) Fit the regression

$$E(I(C = c) \mid X = x) = g_c(x;\theta),$$

allocate to class with largest $g_c(x;\widehat{\theta})$, equivalent to the class with nearest 'target'
$\boldsymbol{t}_c = (0,\dots,0,1,0,\dots)$ to $\boldsymbol{g}(x;\widehat{\theta})$.
[sometimes known as *'minimum (mean) square classifier'*]

(c) Dietterich & Bakiri (1991, 1995) have suggested choosing higher-dimensional targets which are well-spaced in $m$ dimensions for $m > K - 1$.

## 4 'Plug-in' *vs* 'Predictive'

Statistical decision theory (Ripley, 1996, Section 2.4) tells us that as we don't know $P(C = c \mid X = x)$, but should use our best estimate $P(C = c \mid\ X = x, \mathcal{T})$, where $\mathcal{T}$ denotes the training data. This is called the *predictive* approach, whereas using $P(C = c \mid X = x; \widehat{\theta})$ is called the *plug-in* approach.

We have

$$\begin{aligned} P(c \mid x, \mathcal{T}) &= \int P(c \mid x;\theta) p(\theta \mid \mathcal{T}, x)\,\mathrm{d}\theta \\ p(\theta \mid \mathcal{T}, x) &\propto \prod_{i=1}^{n} P(c_i \mid x_i;\theta) p(x_i;\theta)\ p(\theta) p(x;\theta) \end{aligned}$$

and normally assume that $p(x;\theta)$ does not in fact depend on $\theta$ (and hence $p(\theta \mid \mathcal{T}, x)$ does not depend on $x$).

There will be appreciable differences only if $p(c \mid x;\theta)$ is not highly peaked *and* $p(c \mid x;\theta)$ is not linear around the posterior mode for $\theta$. The differences are small in linear discriminant and logistic discrimination, but can be large in quadratic discriminant analysis (Aitchison & Dunsmore, 1975; Ripley, 1996) and non-linear logistic methods.

The predictive approach has been know for many years, has two texts (Aitchison & Dunsmore, 1975; Geisser, 1993) but seems remarkably little known.

## 5 Data on Cushing's Syndrome

Data on diagnostic tests on patients with Cushing's syndrome, a hypertensive disorder associated with over-secretion of cortisol by the adrenal gland. The data come from Aitchison & Dunsmore (1975, Tables 11.1–3).

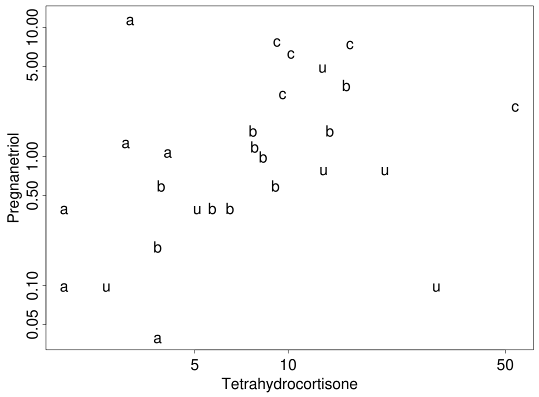



Figure 8: The Cushing's syndrome data.

This dataset has three recognised types of the syndrome represented as a, b, c. (These encode 'adenoma', 'bilateral hyperplasia' and 'carcinoma'.) The observations are urinary excretion rates (mg/24h) of the tetrahydrocortisone and pregnanetriol, and are considered on log scale. Those marked u are future examples to be classified. There are few examples in few dimensions but the ratio of examples to dimensions is quite typical.

Neural networks were fitted in the 'logistic discrimination' framework, by maximum likelihood with weight-decay. Soem fits are shown in figures 9 and 10. Once a moderate number of hidden units are introduced (enough to fit the sort of region we would fit 'by eye') here are lots of local minima. Averaging across 20 fits is quite successful (figure 10).

Can we do better? Yes, a little, *if* we can do the integration. We currently break the integral up into regions about the local maxima of the posterior density, use a Gibbs sampler to sample around each peak and to correct a local Gaussian approximation (Ripley, 1994b), and to find a 'Bayes factor' for each region. By a similar process we can find Bayes factors over number of units and so on (Ripley, 1995).

There have been other approximate approaches to the integration over the network parameters. MacKay's (1992) so-called 'evidence' approach is a version of ML-II empirical Bayes. Neal (1996) uses MCMC methods, but without checking that they converge. (Our experiments suggest that there is great difficulty in adequately covering the parameter space by simple MCMC methods.)

## 6 A Forensic Example

The data in this problem are from 214 fragments of glass collected at scenes of crimes. Each has a measured refractive index and composition (weight percent of oxides of Na, Mg, Al, Si, K, Ca, Ba and Fe). They were grouped as window float glass (70), window non-float glass (76), vehicle window glass (17) and other (containers, tableware, headlamps) (22).

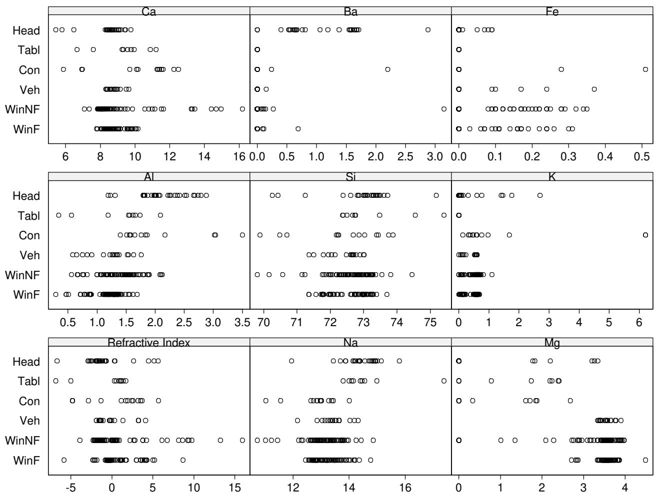



Figure 11: Trellis plots of the forensic glass data.

Our first experiments (Ripley, 1994b) omitted headlamp glass and used 89 examples to train and 96 to test. The results results on the test set were.

| methods | error rates % | |
|---|---|---|
| linear discriminant | 41 | 22 |
| nearest neighbour | 26 | 17 |
| neural network with 2 hidden units | 38 | 16 |
| ditto, averaged over fits | 38 | 14 |
| neural network with 6 hidden units | 33 | 15 |
| ditto, averaged over fits | 28 | 12 |
| classification tree | 28 | 15 |

In the second column of error rates, confusion between the two types of window glass is allowed so the classification is into house window or vehicle window or other. This shows that for nets with enough complexity the predictive approach has marked advantage. (Neal, 1996, discusses the MCMC approach in this problem.)

On a 10-fold cross-validation experiment (Ripley, 1996)

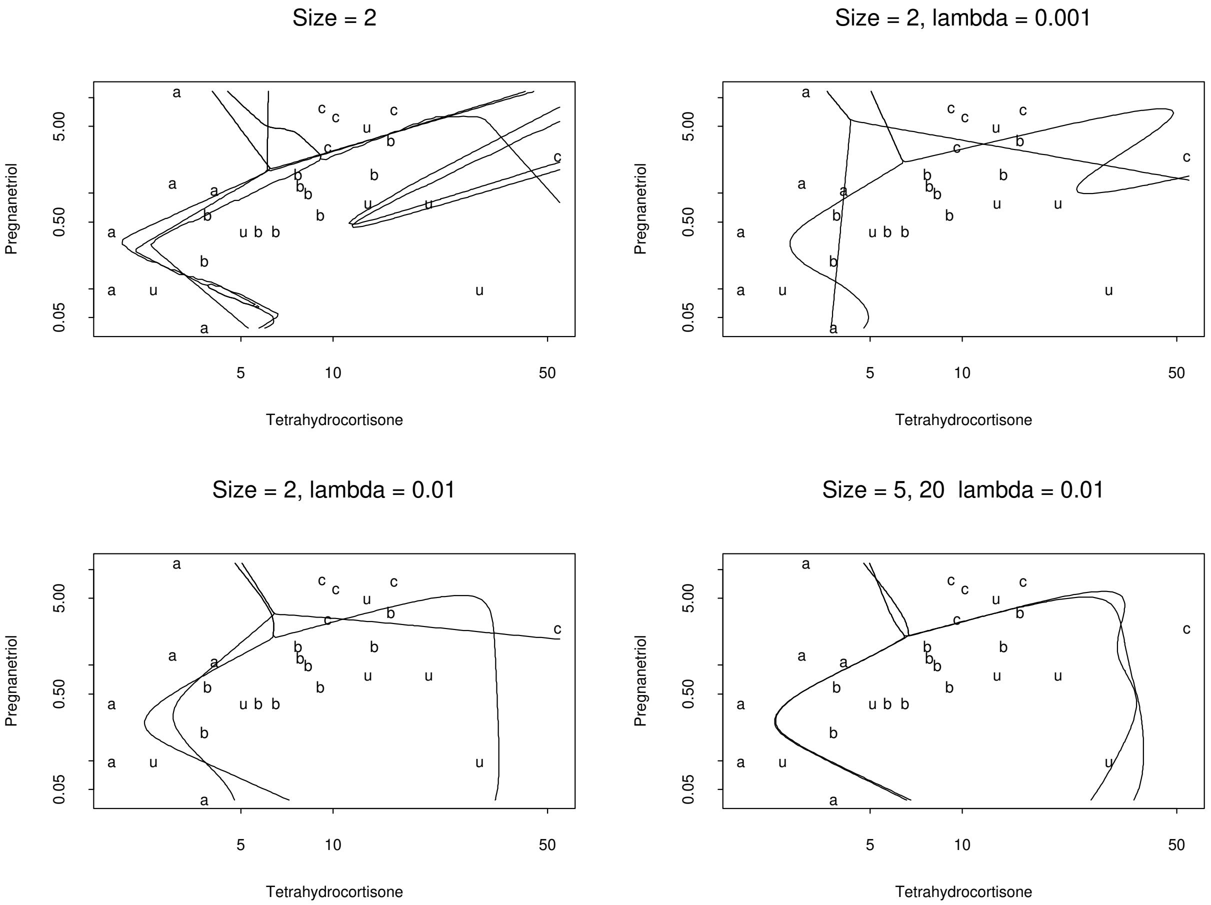



Figure 9: Neural network fits to the Cushing's syndrome data. Two local minima are shown for each of the first three panels. The fit in the lower right panel has 20 hidden units, but is controlled almost entirely by the weight decay. Note that most of the fits on the top row are perfect.

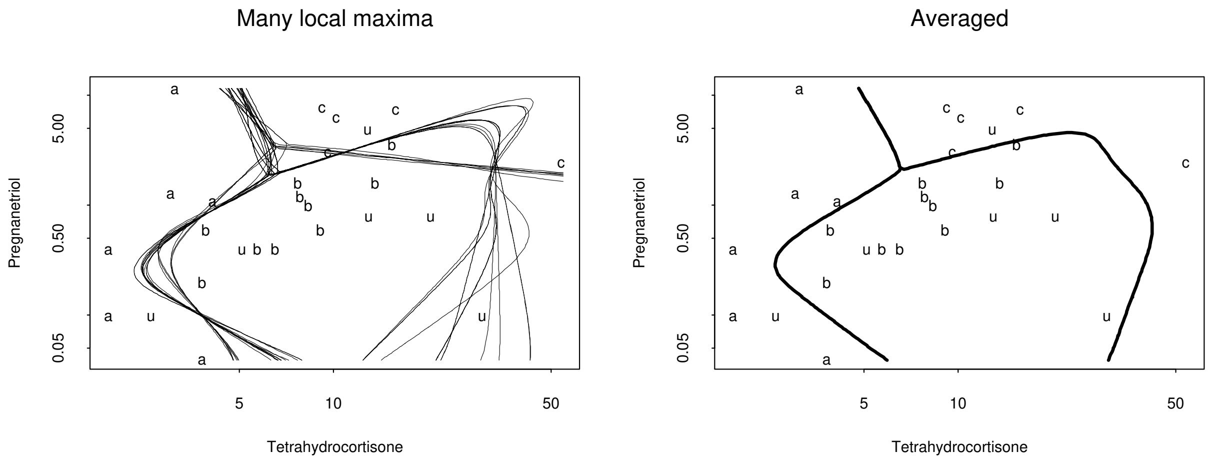



Figure 10: Neural networks of size 3 with $\lambda = 0.01$ applied to the Cushing's syndrome data. The left panel shows many local minima found from 25 randomly chosen starting points. The right panel shows the decision regions found by averaging the predictions $p(c \mid x)$ of the 25 fits.

we found error rates (%) of

| $\lambda$ | # (hidden units) 2 | 4 | 8 |
|---|---|---|---|
| 0.0001 | 30.8 | 23.8 | 27.1 |
| 0.001 | 30.4 | 26.2 | 26.2 |
| 0.01 | 31.8 | 29.9 | 29.9 |

This shows that only a small amount of regularization by weight decay is needed in this dataset; this is related to the near-linear separation of some of the classes. The sharp variation in performance with $\lambda$ and the number of hidden units in this example is unusual, and can be traced to the success in separating the rare types of glass.

Attempts (Ripley, 1996, p. 163) to combine this classifiers with others (especially nearest-neighbour classifiers) produced no performance advantage.

Data and S code for both this example and the previous one are supplied with Venables & Ripley (1997).

## References


Aitchison, J. & Dunsmore, I. R. (1975) *Statistical Prediction Analysis.* Cambridge: Cambridge University Press.

Barron, A. R. (1993) Universal approximation bounds for superpositions of a sigmoid function. *IEEE Transactions on Information Theory* **39**, 930–945.

Barron, A. R. (1994) Approximation and estimation bounds for artificial neural networks. *Machine Learning* **14**, 115–133.

Bates, D. M. & Watts, D. G. (1988) *Nonlinear Regression Analysis and its Applications.* New York: Wiley.

Bishop, C. M. (1995) *Neural Networks for Pattern Recognition.* Oxford: Clarendon Press.

Cheng, B, and Titterington, D. M. (1994) Neural networks: a review from a statistical perspective (with discussion). *Statistical Science* **9**, 2–54.

Cherkassky, V., Friedman, J. H. & Wechsler, H. (eds) (1994) *From Statistics to Neural Networks. Theory and Pattern Recognition Applications.* Berlin: Springer.

Geisser, S. (1993) *Predictive Inference: An Introduction.* New York: Chapman & Hall.

Girosi, F. & Anzellotti, G. (1993) Rates of convergence for radial basis functions and neural networks. In *Artificial Neural Networks for Speech and Vision* ed R. J. Mammone, pp. 97–114. London: Chapman & Hall.

Hardy, R. L. (1971) Multiquadric equations of topography and other irregular surfaces. *Journal of Geophysical Research* **76**, 1906–1915.

Hardy, R. L. (1990) Theory and applications of the multiquadric-biharmonic method: 20 years of discovery 1968–1988. *Computers and Mathematics with Applications* **19**, 163–208.

Haykin, S. (1994) *Neural Networks. A Comprehensive Foundation.* New York: Macmillan College Publishing.

Hertz, J., Krogh, A. & Palmer, R. G. (1991) *Introduction to the Theory of Neural Computation.* Redwood City, CA: Addison-Wesley.

Kung, S. Y. (1993) *Digital Neural Networks.* Englewood Cliffs, NJ: Prentice-Hall.

MacKay, D. J. C. (1992) A practical Bayesian framework for backprop networks. *Neural Computation* **4**, 448–472.

Michie, D., Spiegelhalter, D. J. & Taylor, C. C. (eds) (1994) *Machine Learning, Neural and Statistical Classification.* New York: Ellis Horwood.

Neal, R. M. (1996) *Bayesian Learning for Neural Networks.* Lecture Notes in Statistics **118**. New York:Springer.

Powell, M. J. D. (1987) Radial basis functions for multivariable interpolation: a review. In *Algorithms for Approximation*, eds J. C. Mason & M. G. Cox, pp. 143–167. Oxford: Clarendon Press.

Ripley, B. D. (1993) Statistical aspects of neural networks. In *Networks and Chaos—Statistical and Probabilistic Aspects*, eds O. E. Barndorff-Nielsen, J. L. Jensen & W. S. Kendall, pp. 40–123. London: Chapman & Hall.

Ripley, B. D. (1994a) Neural networks and related methods for classification (with discussion). *J. Roy. Statist. Soc. B* **56**, 409–456.

Ripley, B. D. (1994b) Flexible non-linear approaches to classification. In Cherkassky *et al.* (1994), pp. 105–126.

Ripley, B. D. (1995) Statistical ideas for selecting network architectures. In *Neural Networks: Artificial Intelligence and Industrial Applications*, eds B. Kappen & S. Gielen, pp. 183–190. London: Springer.

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

Weiss, S. M. and Kulikowski, C. A. (1991) *Computer Systems that Learn:Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems.* San Mateo: Morgan Kaufmann.

Venables, W.N. and Ripley, B.D. (1997) *Modern Applied Statistics with S-Plus.* Second edition. New York: Springer.

Wahba, G. & Wold, S. (1975) A completely automatic French curve. *Communications in Statistics* **4**, 1–17.