development for cognitive science. Such interactions will lead to a deeper understanding of the interpretation and learning tasks, and may ultimately help us to address other cognitive tasks, perhaps including creative thinking and scientific discovery, as well.

ACKNOWLEDGMENTS

Support for the preparation of this article was provided by NIMH Project Program Grant MH47566-03 and by NIMH Research Scientist Award MH00385-13.

Comment

B. D. Ripley

Bing Cheng and Mike Titterington have reviewed many of the areas of neural networks; their paper overlaps the flood of books on the subject. I also recommend Weiss and Kulikowski (1991) (Segre and Gordon, 1993, provide an informative review) and Gallant (1993) for their wider perspective and Wasserman (1993) for coverage of recent topics. My own review article, Ripley (1993a), covers this and many of the cognate areas as the authors comment. The five volumes of the NIPS proceedings (*Advances in Neural Information Processing Systems*, 1989–1993, various editors) provide a very wide-ranging overview of highly-selected papers. Much of the latest work is available electronically from the ftp archive at `archive.cis.ohio-state.edu` in directory `pub/neuroprose`.

At the time I received this paper to discuss, I had recently attended a NATO Advanced Study Institute on *From Statistics to Neural Networks* (whose proceedings will appear as Cherkassky, Friedman and Wechsler, 1994), which despite the direction of the title revealed that current thoughts in neural networks are not to subsume statistics in neural networks but vice versa. Many researchers in neural networks are becoming aware of the statistical issues in what they do and of relevant work by statisticians which encourages fruitful discussions.

Cheng and Titterington concentrate on similarities between statistical and neural network methods. I feel the differences are more revealing as they indicate room for improvement on at least one side. However, I believe the most important issues to be those of practice which are almost ignored in the paper. Before I turn to those, there are two points I wish to attempt to clarify.

B. D. Ripley is Professor of Applied Statistics, University of Oxford, 1 South Parks Road, Oxford OX1 3TG, United Kingdom. This comment was written while on leave at the Isaac Newton Institute for Mathematical Sciences, Cambridge, United Kingdom.

1. PROJECTION-PURSUIT REGRESSION

The connection between multilayer perceptrons (MLPs) and projection-pursuit regression (PPR) is much deeper than the authors appear to suggest. Other empirical comparisons (apart from my own cited in the paper) are given by Hwang et al. (1992a,b, 1993), and Barron and Barron (1988) viewed PPR from a network viewpoint. In the authors' notation PPR is

$$y_i = w_{0i} + \sum_k \gamma_i \psi_k (x^T v_k),$$

where I have allowed for multiple outputs. An MLP with linear output units is the special case of logistic ψ_k; of course both PPRs and MLPs can be given nonlinear output units. Since we can approximate any continuous ψ_k of compact support uniformly by a step function and can approximate (nonuniformly) a step function by a logistic, we can approximate ψ_k uniformly by a sum of logistics. This fact plus the (elementary) approximation result for PPR of Diaconis and Shahshahani (1984) gives the approximation results of Cybenko and others. There is a version of Barron's L_2 result for PPR by Zhao and Atkeson (1992). (This point of view, approximating ψ_k by a simple neural net of one input, corresponds to organized weight-sharing between input-to-hidden-unit weights for groups of units, a sensible procedure in its own right.)

These results suggest that the approximation capabilities of MLPs and PPR are very similar (suggesting an affirmative partial answer to the question in Section 7). However, PPR will have an advantage when there are many inputs, only a few combinations of which are relevant, in making better use of each projection and hence fewer projections and parameters. My suspicion is that this is commonly the case.

2. HANDWRITTEN DIGIT RECOGNITION

The literature on handwriting recognition, especially studies of Zip-codes, is much misquoted and I suspect much misunderstood. Many of the best methods for handwriting recognition depend on choosing good features, and the lower levels of the Le Cun system can be thought of as feature extraction not classification and were originally optimized by hand. The actual results are often stated confusingly (including in Le Cun et al., 1990). There is a training set of size 9709, 7291 of which are handwritten, and a test set of 2007 handwritten characters plus some others. According to Vapnik (1992) the test-set error rate for the Le Cun system is 5.1% (102/2007) (but the 1990 variant would appear to achieve 4.6% (92/2007)) against 2.5% for humans and 3.3% for the best automated system to date. Undoubtedly, the automated systems have been optimized for this particular dataset, so these rates may be a little optimistic.

Later workers have suggested that the handcrafting is not necessary and report similar results from simpler applications of neural networks: Knerr, Personnaz and Dreyfus (1992) and Martin and Pitman (1990, 1991). Grother and Candella (1993) report best results for the *probabilistic neural network* of Specht (1990), that is kernel discriminant analysis, using up to 64 principal components of the 128^2 image data as input features. These are all general purpose methods, at least as much so as penalized discriminant analysis (PDA) and achieve error rates of around 2.5%. Against this, the value of PDA, with an error rate of 8.2%, is surely overstated.

This example shows the difficulty of quoting error rates without reference to the Bayes risk. The latter can often be estimated (Fukunaga, 1990; Ripley, 1994b); but in this case, it must be close to the error rate achieved by humans. It is also potentially confusing to quote per-digit error rates when the task depends on correctly reading whole Zip-codes. That task has some redundancy (not all possible Zip-codes are valid nor equally probable) and high correlation in the errors for the separate digits. The residual error rate contains both segmentation errors in isolating the digits and plain errors (wrongly labelled digits). There is substantial interwriter variability, and careful studies (such as that of Grother & Candella) use different writers for the training and test sets.

3. OPTIMIZATION IN FITTING MLPs

The comments in Sections 4.2.2 and 4.3.2 hide a series of very important practical points. Some authors argue that the point of the back-propagation gradient-descent algorithm is *not* to minimize $E(W)$ since doing so will lead to over-fitting. The regularization approach is to add a term to penalize rough functions (such as weight decay) and so change the objective to a function we really do want to minimize. Other people believe in stopping early as a means of regularization, although why travelling along a path in the wrong direction to the nearest point to a goal is thought a good procedure beats me. (It also occurs in statistical approaches to tomography, e.g., Vardi and Lee, 1993.)

What is clear is that no experienced worker attempts to minimize $E(W)$ alone, and this makes comparisons of methods difficult. A typical approach is to stop when the error measure on a validation set starts to rise. This has a number of difficulties:

- To repeat the point, there is no guarantee that the path taken is sensible.
- In my experience, the error on the validation set often rises for a while then falls dramatically before rising again; therefore, it is impossible to know that the best point on the path has yet been reached.
- The use of a validation set wastes data, and I suspect that often the test set is used. One example, in a textbook, is Thornton (1992, p. 199).

A further difficulty is the prevalence of local minima, which are much more common than comments in the literature (e.g., Thornton, 1992, Section 13.6) suggest—it needs careful work to discover many of the minima of the error surface.

Schiffmann, Joast and Werner (1992) and Jervis and Fitzgerald (1993) report studies of a wide range of optimization techniques on a narrow range of problems, and both review the literature. Their conclusions differ, and their experience differs from my own. It does seem that the more sophisticated methods (such as quasi-Newton and conjugate gradients) do best in hard optimization problems, often dramatically so (e.g., Grother and Candella, 1993), but can be beaten by on-line gradient descent methods on simpler tasks.

The back-propagation algorithm can be extended to compute second derivatives in some or all directions (Bishop, 1992; Buntine and Weigend, 1993; Pearlmutter, 1994). Interesting developments in this area include RProp (Riedmiller and Braun, 1992) and scaled conjugate gradients (Møller, 1993) which can make use of Pearlmutter's techniques.

It is worth noting that in the Bayesian approach the effort of minimization is redirected to integration over the weights, either by a saddlepoint approximation or by Monte-Carlo methods (e.g., Neal, 1993). (We will almost never be interested in the

weights *per se* despite the emphasis of Section 4.3.5.) It is not yet clear how much effort is needed to do the integration well.

4. METHODS FOR CLASSIFICATION

The authors mention Hastie, Tibshirani and Buja's FDA in Sections 4.3.1 and 7. These authors and I studied Breiman and Ihaka's unpublished 1984 paper to see if such simple results had a simple explanation and rederived the results via canonical correlations. My version will appear in Ripley (1993b, 1994b) and in detail in Ripley and Hjort (1994).

For two normally distributed classes with a common covariance matrix, it is well known that the sample linear discriminant (LDF) is more efficient than logistic discrimination since it uses full rather than conditional maximum likelihood and that the LDF can be found by regression up to the additive constant.

We can think of the linear regression as the best linear approximation to the posterior probabilities and extend this to more than two classes. As a principle of classifier design, this has been used (Duda and Hart, 1973; Devijver and Kittler, 1982; Fukunaga, 1990) under the name of *minimum (mean) square error* classifiers. Unlike the linear discriminant, this procedure classifies by the nearest target or equivalently the largest component of a regression for each class indicator. What Breiman and Ihaka showed is that the regressions span the same space as the canonical variates and that the linear discriminant classifies by choosing the nearest target in a non-Euclidean metric in that space.

Neural networks (at least, MLPs and RBFs) are nonlinear regressions. This suggests a number of ways to use them for classification:

- (What Hastie, Tibshirani and Buja, 1992, called FDA). Regress the class indicators on the input variables and use LDA in the space of fitted values. Equivalently, encode the classes in scores and regress the scores on the inputs.
- Use the functions in a nonlinear model for the log posterior probabilities. This is sometimes known as *softmax* in this field and fitted via maximum likelihood and is possibly penalized by, say, weight decay.
- Use the functions for separate nonlinear logistic models for each class *versus* the rest, as in an MLP with logistic output units. Although apparently less sensible than the previous method, this is by far the most commonly used, for example, in the Le Cun study.
- Choose well-separated scores for the classes and regress on the inputs (Dietterich and Bakiri, 1991).

The authors appear to prefer the first method, but they probably have no practical experience. I have found a number of difficulties, over many experiments, that stem from the need to estimate the within-class covariance in the space of fitted values. For fits from nonlinear regressions (including MLPs, RBFs, MARS and projection pursuit regression) the covariance matrix can be dominated by outliers; and even with robust estimation, it can be insufficiently well determined. My current preference is for the second approach, but this raises problems for techniques such as MARS that are tailored to least-squares fitting. My impression is that how the flexible family of functions is used is much more important than which family is chosen.

5. WHAT CAN NEURAL NETWORKS ACHIEVE?

It is no accident that all the real examples Cheng and Titterington chose are classification problems; in my reading, these form over 90% of the applications with regression techniques being used in time series (Weigend and Gershenfeld, 1993) and control (Miller, Sutton and Werbos, 1990). Great advances have been claimed for neural networks, but more careful studies have shown that in many of the cited examples statistical methods can do as well or even much better. (For NETtalk, Wolpert, 1990; for digit recognition, Grother and Candela, 1993; for the sonar problem of Gorman and Sejnowski, 1988a, b; Ripley, 1994a.) Often linear methods or k-nearest neighbour methods, used carefully, will do as well as neural networks.

There should, though, be a place for methods between the linear parametric methods and wholly nonparametric methods for highly-parametrized methods such as MLPs, RBFs, MARS and projection pursuit regression, especially in problems with significantly curved structure and relatively few data points.

One thing clients often require is to be able to *understand* the classifier. This is difficult with black-box systems such as neural networks and is often claimed as an advantage of machine-learning systems such as tree- and rule-induction systems (Quinlan, 1993; Thornton, 1992). This may be true if there is a simple true classifier. In other cases, the true relationship between classes appears to be too complicated to be perceived easily (such as the forensic glass example in Ripley, 1994a, b). Humans often find rules easiest to comprehend; and any classifier can be approximated by a rule system, for example, by generating examples from it and inducing rules from these (as in Gallant, 1993 or Quinlan, 1993).

Issues of choosing model complexity and assessing performance and "generalization" (Section 4.3.4) are among the most important open questions. There is some evidence that methods such as cross-validation and AIC are too "local" to fully assess the variability of very flexible methods; therefore some of the assessed benefits of nonlinear methods may be illusory. [On "generalization", Haussler (1992) is a far-reaching extension of the ideas of VCdim to which statisticians, especially David Pollard and Luc Devroye, have contributed; and Anthony and Biggs (1992) is an introductory text on the seminal ideas of Blumer et al., 1989.]

One thing statisticians can contribute to the debate is experience in careful use of sophisticated nonlinear methods. Software is readily available, including in S, and I would encourage statisticians to experiment rather than quote inadequately designed propaganda studies.

To end on a positive note, some very impressive applied statistics is being done using neural networks, and the explosive growth of the subject has opened the eyes of some statisticians (including myself) to the complexity of problems that may be fruitfully attacked by nonlinear methods. I and others have been particularly impressed by some work of my Oxford Engineering Science colleague, Lionel Tarassenko, on analyzing sleep EEG data using both Kohonen nets and radial basis functions to detect structure and anomalous signals (Roberts and Tarassenko, 1993, 1994).

Comment

Robert Tibshirani

Cheng and Titterington's paper is a scholarly overview of the field of neural networks. It should raise the statisticians' awareness of this interesting and important field. One of the authors' objectives was to encourage cross-disciplinary research between neural network researchers and statisticians. Here at the University of Toronto, I have been collaborating informally with Geoffrey Hinton of the Computer Science department, and I think that this collaboration has been fruitful for both of us.

First I would like to make a general point drawing a distinction between statistics and neural networks:

Statisticians tend to work with more interpretable models, since measuring the effects of individual input variables, rather than prediction, is often the purpose of the analysis.

Having said that, there is still much that one field can learn from the other. I will briefly summarize some of the main points:

WHAT THE STATISTICIAN CAN LEARN FROM NEURAL NETWORK RESEARCHERS

1. We should worry less about statistical optimality and more about finding methods that work, especially with large data sets.
2. We should tackle difficult real data problems like some of those addressed by neural network researchers, like character and speech recognition and DNA structure prediction. As John Tukey has said, it is often better to get an approximate solution to a real problem than an exact solution to an oversimplified one.
3. Models with very large numbers of parameters can be useful for prediction, especially for large data sets and problems exhibiting high signal-to-noise ratios.
4. Modelling linear combinations of input variables can be a very effective approach because it provides both feature extraction and dimension reduction.
5. Iterative, nongreedy fitting algorithms (like steepest descent with a learning rate) can help to avoid overfitting in models with large numbers of parameters.
6. We (statisticians) should sell ourselves better.

WHAT THE NEURAL NETWORK RESEARCHER CAN LEARN FROM STATISTICIANS

1. They should worry more about statistical optimality or at least about the statistical properties of methods.
2. They should spend more effort comparing their methods to simpler statistical approaches. They will be surprised how often linear regression performs as well as a multilayered percep-

Robert Tibshirani is Associate Professor, Department of Preventive Medicine and Biostatistics, University of Toronto, 12 Queens Park, Toronto, Ontario M5S 1A8, Canada.