USGS
science for a changing world

Cost-Benefit Analysis of Computer Resources for Machine Learning

Open-File Report 2007–1398

U.S. Department of the Interior
U.S. Geological Survey

THIS PAGE IS INTENTIONALLY BLANK

Cost-Benefit Analysis of Computer Resources for Machine Learning

By Richard A. Champion, Jr.

Open-File Report 2007–1398

U.S. Department of the Interior
U.S. Geological Survey

U.S. Department of the Interior
DIRK KEMPTHORNE, Secretary

U.S. Geological Survey
Mark D. Myers, Director

U.S. Geological Survey, Reston, Virginia: 2007

For product and ordering information:
World Wide Web: http://www.usgs.gov/pubprod
Telephone: 1-888-ASK-USGS

For more information on the USGS--the Federal source for science about the Earth, its natural and living resources, natural hazards, and the environment:
World Wide Web: http://www.usgs.gov
Telephone: 1-888-ASK-USGS

Suggested citation:
Champion, Richard, Jr., 2007, Cost-benefit analysis of computer resources for machine learning: U.S. Geological Survey Open-File Report 2007–1398, 7 p.

Contents

Abstract....1
Introduction....1
A Cost-Benefit Experiment Using Geospatial Correlation....2
Discussion and Future Experiments....4
    Geometry of the Data Cloud....4
    Bayesian Samples and Stopping Points....5
    Retraining....5
    Multivariate Experiments....6
Acknowledgements....6
References Cited....6

Figures

1. Normalized road and dasymetric density across the San Francisco Bay Area. Road density is a measure of distance to the nearest road for each pixel. Normalization scales this to the range 0 to 1. Dasymetric density uses census-block and land-cover information to estimate population density per 30-m pixel. Population density values near the high end of the scale (near 1 person per 30-m pixel) suggest artifacts generated from inaccuracies in the land-cover information....3
2. Calibration time as a function of the number of training points. The training time is approximately quadratic (indicated by the exponent 2.19) and explains more than 98 percent of the variance in the performance data....3
3. Population density predicted from a probabilistic neural network using normalized road density and training-set sizes of 1,000 and 50,000 points. The average of the curves suggests how the goodness-of-fit varies with training sets of intermediate sizes. The blips in the curves for normalized road density greater than about 0.6 are likely due to noise in the data. For these data a large increase in the number of training points does not lead to a large improvement in goodness of fit....4
4. Scatterplot showing the geometrically uneven distribution of data points. The clustering of data near the origin indicates that the highest population densities are found in the areas containing the highest density of roads....5

THIS PAGE IS INTENTIONALLY BLANK

Cost-Benefit Analysis of Computer Resources for Machine Learning

By Richard A. Champion, Jr.

Abstract

Machine learning describes pattern-recognition algorithms—in this case, probabilistic neural networks (PNNs). These can be computationally intensive, in part because of the nonlinear optimizer, a numerical process that calibrates the PNN by minimizing a sum of squared errors. This report suggests efficiencies that are expressed as cost and benefit. The cost is computer time needed to calibrate the PNN, and the benefit is goodness-of-fit, how well the PNN learns the pattern in the data. There may be a point of diminishing returns where a further expenditure of computer resources does not produce additional benefits. Sampling is suggested as a cost-reduction strategy. One consideration is how many points to select for calibration and another is the geometric distribution of the points. The data points may be nonuniformly distributed across space, so that sampling at some locations provides additional benefit while sampling at other locations does not. A stratified sampling strategy can be designed to select more points in regions where they reduce the calibration error and fewer points in regions where they do not. Goodness-of-fit tests ensure that the sampling does not introduce bias. This approach is illustrated by statistical experiments for computing correlations between measures of roadless area and population density for the San Francisco Bay Area. The alternative to training efficiencies is to rely on high-performance computer systems. These may require specialized programming and algorithms that are optimized for parallel performance.

Introduction

Patterns in a GIS data layer, in a remotely sensed image, or on the landscape can be visually obvious to the human eye. Machine-learning (Poggio and Smale, 2003) is a mathematical discipline that describes generalized patterns so that computers can mimic the human skill of pattern recognition. This report considers one type pf machine-learning: probabilistic neural networks (PNNs) (Masters, 1993, 1995). PNNs have the advantage (Ripley, 1996) over alternative forms of neural networks, such as Feed Forward Neural Networks (FFNs), of being relatively easy to train. Machine-learning has a demonstrated value in recognizing complex patterns in large environmental data sets. Conrads and others (2002) and Roehl and Conrads (1999) describe the use of neural networks and data mining for the modeling of estuary and river systems. Risley and others (2003) describe a combination of unsupervised and supervised learning techniques to model water temperature in streams over an 80,000 square kilometer region in western Oregon.

Practical considerations in applying machine learning are whether the programs can run on a desktop computer system, or whether a high-performance computer system is required. The system used in this study for performance measurements has a Pentium 4 dual-core processor (3.20 and 3.19 GHz) with 1.00 GByte of RAM. For this system there are formulations of machine-learning that can expend tens and even more hours of calibration time. This report suggests alternatives for efficient training, improving the utilization of computer resources by requiring the machine-learning algorithm to work no harder than it has to in order to learn and generalize the pattern in the data. This is expressed in terms of cost-benefit measures, where cost is execution time and the benefit is framed in terms of statistical measures of goodness-of-fit. The PNN software for these experiments is from Masters (1995).

Part of the computational demand of machine-learning is due to a numerical process called optimization. Numerical optimizers are used across a wide variety of machine-learning algorithms (Bishop, 2006; Cucker and Smale, 2001; Hastie and others, 2001). Using a training set, the optimizer teaches the machine-learning algorithm the data pattern by finding parameters that describe the minimum of an error function. This is analogous to finding the best least-squares estimate as in ordinary regression, except that in this case the error function can be highly nonlinear. The computational burden to minimize the error function tends to increase dramatically with the number of points in the training set. It may be that a small set of training data can minimize the error function as well as a larger training set. If so, then the algorithm can learn the pattern in the data using a smaller rather than a larger training set and in a shorter rather than in a longer time. The numerical experiments below quantify, for a particular example, computation time as a function of sample size. The experiments also suggest how to evaluate convergence—a measure of when a calibration set contains enough information to give a reliable fit.

An alternative is a more powerful computer system, either a sequential system with a faster processor or a system with parallel architecture. In sequential computer architectures, instructions are executed one at a time in a single logic stream until the program finishes. In parallel computer architectures, instructions split into parallel logic streams that are independently executed. A variation is message passing, where, at programmer-defined intervals, the separate logic streams share information. See Pacheco (1997) for a discussion of Message Passing Interface (MPI). Parallel programming paradigms may require restructuring of serial code, translation from one computer language to another, or the development of parallel algorithms. An ideal programming paradigm would allow serial code to take advantage of a parallel processor with minimal program reorganization. The reality is that extensive programming may be necessary to make best use of the parallel system. See Mattson and others (2004) for parallel system-programming paradigms.

The optimization algorithm for the machine-learning used here is based on a mathematical algorithm called simulated annealing (Albright, 2007). The C++ source code for the optimizer is taken from Press and others (1992). The optimizer searches for a global minimum of an error function in multidimensional space. Ordinary Euclidian space has three or more dimensions or coordinates per point. Multidimensional space can have any finite number of dimensions. If there are five independent variables, then the error function would be a hypersurface in this space. Because searches in higher dimensional spaces tend to be slower than searches in spaces of lower dimensions, the computer time to calibrate a machine-learning algorithm increases with the number of variables. Also, there are quirks in the geometry of the error surface (Masters, 1993, 1995) that lead to an increase in execution time. Calculus techniques (Newton's and conjugate gradient methods) require strong assumptions about the geometry of the error surface and the location of the minimum. These methods also assume a specified starting point for iteration that is near the minimum, where "nearness" is intentionally left vaguely defined. In simulated annealing, the optimizer randomizes the initial points and applies the calculus techniques, given each randomized starting point. The optimizer finally accepts as the global (best overall) minimum the result that is at least as good as the rest. This requires extensive iteration and is computationally intensive. The use of a parallel system for machine-learning would require a parallel version of the optimization algorithm (Aarts and Korst, 1989).

A Cost-Benefit Experiment Using Geospatial Correlation

This section describes a statistical experiment to investigate efficiencies in the calibration and use of PNNs using an example. Efficiency is described as cost and benefit, where cost is training time and benefits are statistical measures of goodness-of-fit. For PNNs the calibration time and the generalization time—the time for the PNN to predict outcomes from data that it has not yet seen—are related. Reducing the calibration time also reduces the time required by the PNN for modeling. The experiment is based on geospatial correlation. Because more people tend to live in areas where there are more roads and fewer people tend to live where there are fewer roads, the approximate quantitative relationship between population and roadless density can be guessed in advance. This allows the experiment to primarily consider the cost-benefit aspects of training. The data sets are of similar vintage and quality, and are known to contain noise (fig. 1). This provides an opportunity to calibrate a PNN in the presence of noise. The points in the data set tend to cluster near the origin and are thin elsewhere. This provides an opportunity to consider how to design algorithms for stratified sampling of training points.

Roadless density is a measure of how road-free an area is (Watts and others, 2007). The roadless data (U.S. Geological Survey, The Road Indicator NORM ED Data sets) were taken from the Northern California data set, clipped to the San Francisco Bay Area, and normalized from 0 to 1. The original metric is Euclidean distance, in meters, to the nearest road. The population density is derived by reallocating population from census-block groups to individual pixels by a process called dasymetric mapping (Sleeter, 2004). The dasymetric data are from 1990 and are truncated to the range 0 to 1. This range was selected by considering the average population density for San Francisco (U.S. Census Bureau, 2007), which for 1990 and 2000 is about 0.2 persons per 30-meter pixel. The value of 1 person per pixel was selected as an upper credible bound that keeps the valid data while excluding much of the noise.

To assess the cost of training, the PNN is calibrated using training sets of various sizes. The calibration time is plotted as a function of number of points in the training set, and a functional form is fit through the data. The calibration time appears to be quadratic (fig. 2). This experiment uses about 0.005 to .266 percent of the points in the complete set (size: approximately 7,000 by 7,000 pixels). Increasing this fraction to, say, 10 percent would be significantly more costly for the available computer system.

The question is whether increased cost buys a worthwhile improvement in goodness-of-fit. Figure 3 shows selected population density curves predicted from normalized road densities using the PNN. A prediction curve was computed for each of the sample sizes indicated as a data point in Fig. 2. All curves tracked closely, so figure 3 shows only the averaged curves and the individual curves for 1,000 and 50,000 training points. Increasing the number of training points increases the demand on computer resources and the training time. However, a large increase in the number of training points does not significantly change the predictions, that is, it does not give much improvement in goodness-of-fit, except perhaps to squeeze out some additional noise. There is a precipitous drop in population density as the roadless density increases. The curves are frequently near zero (except for occasional spikes) after the drop. This suggests an actual correlation—not noise alone—on the left side of the plot, with some real effect and noise near the right side.

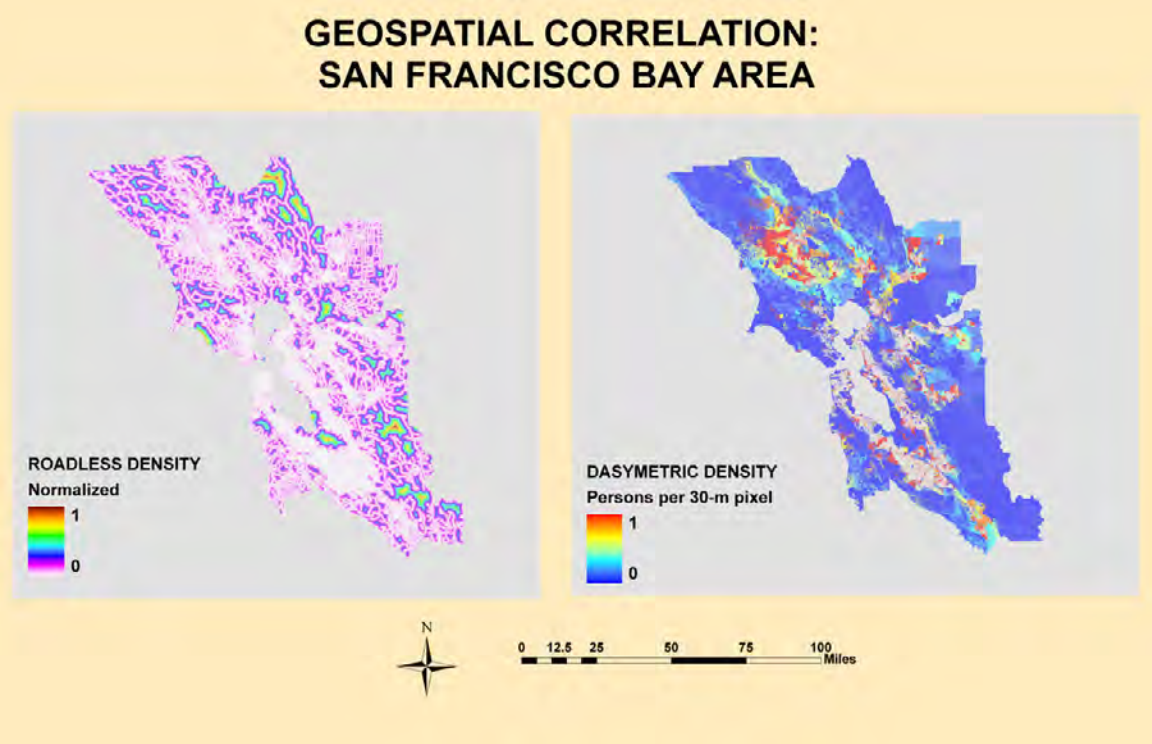

Figure 1. Normalized road and dasymetric density across the San Francisco Bay Area. Road density is a measure of distance to the nearest road for each pixel. Dasymetric density uses census-block and land-cover information to estimate population density per 30-m pixel. Normalization scales values to the range 0 to 1. Population density values near the high end of the scale (near 1 person per 30-m pixel) suggest artifacts generated from inaccuracies in the land-cover information.

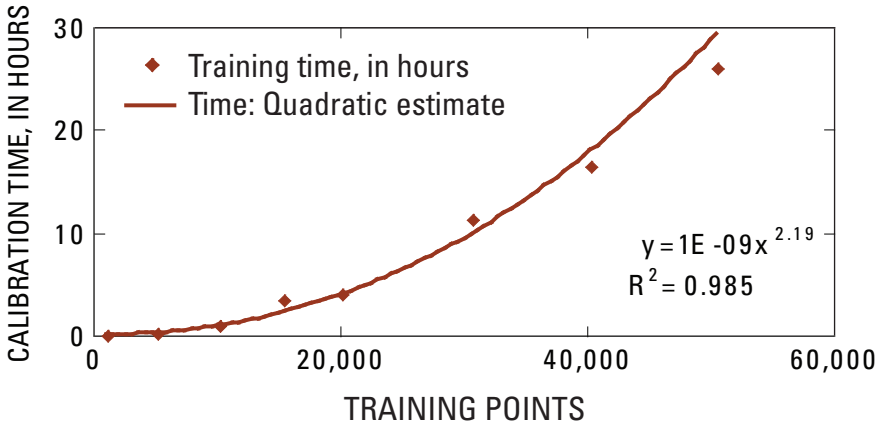

Figure 2. Calibration time as a function of the number of training points. The training time is approximately quadratic (indicated by the exponent 2.19) and explains more than 98 percent of the variance in the performance data.

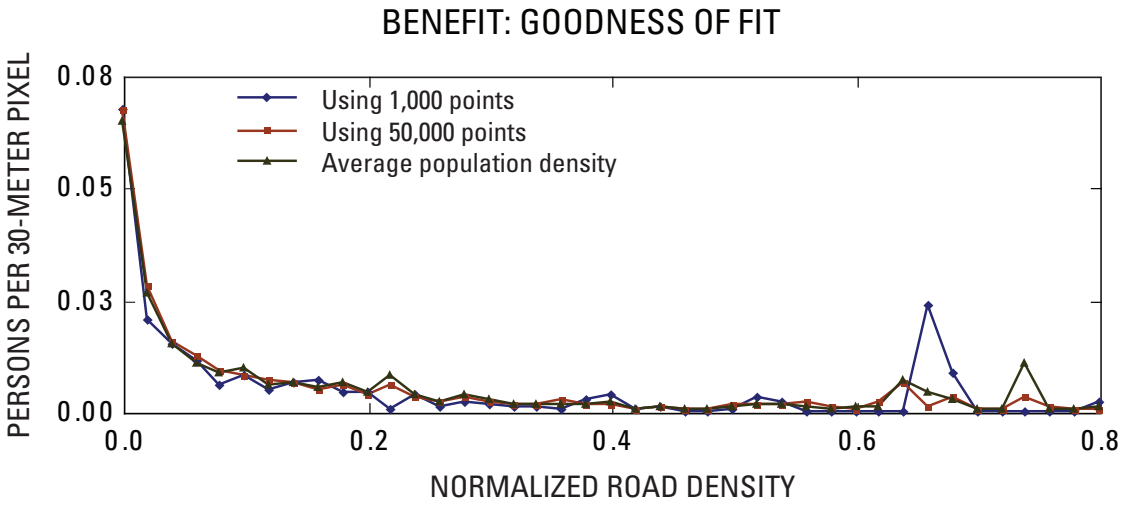

Figure 3. Population density predicted from a probabilistic neural network using normalized road density and training-set sizes of 1,000 and 50,000 points. The average of the curves suggests how the goodness-of-fit varies with training sets of intermediate sizes. The blips in the curves for normalized road density greater than about 0.6 are likely due to noise in the data. For these data a large increase in the number of training points does not lead to a large improvement in goodness of fit.

Discussion and Future Experiments

The statistical experiments have illustrated an approach to the efficient use of computer resources for calibrating and running a neural network model. The approach consists of deriving calibration and convergence curves for training sets of various sizes. The sampling process started with small training sets and worked up to larger training sets. In the course of the analysis, it was shown that the functional form of the calibration curve was, in this case, quadratic (fig. 2). This gives a cost, in terms of calibration time, of including more training points. In the course of the analysis the goodness-of-fit changed with the number of training points (fig. 3). This is the benefit of using more points in the training set. This gives a convergence criterion, which in this case was decided by judgment. The example, geospatial correlation is simple and the conclusion that the two variables—roadless density and population density—are highly correlated is clear from the GIS visualization (fig. 1). Essentially, a small sample of calibration points may be sufficient to properly calibrate a PNN, and the cost-benefit, in terms of computer resources, can be substantial. This example also shows that, even in a simple case, there is a calibration scenario that can be tediously slow on a desktop computer system.

Additional experiments might test techniques for further efficiencies in more realistic and complex problems. This requires a definition of multivariate function and local approximation. A real multivariate function is of the form:

$$f : R^m \rightarrow R^n$$

This is a mathematical way of saying that f is a function of m independent and n dependent variables. In the example of geospatial correlation, the function was univariate with m and n both 1. If m=2 and n=1, then the PNN will fit a data cloud to a surface in three dimensional space. In the univariate (one independent and one dependent variable) case the PNN model is a curve described by connected arcs. This is a local approximation, because the curve is described in pieces. Each additional piece requires new interconnections in the neural network, and in some cases the increase in network complexity will lead to a large increase in run time. The question is how many arcs are needed to properly describe the curve. An insufficient number will give a curve that is overgeneralized, while too many give a network that may require a long run time. Experiments with image data (not shown here) suggest that the run time of the model increases with the number of training points, with the prediction cost apparently linear.

Geometry of the Data Cloud

One efficiency might be gained by devising a sampling technique that pays attention to the geometry of the data cloud. Figure 4 is a scatterplot suggesting the geometric distribution of the data points for normalized road and population densities. Because of the large data volume and the concentration

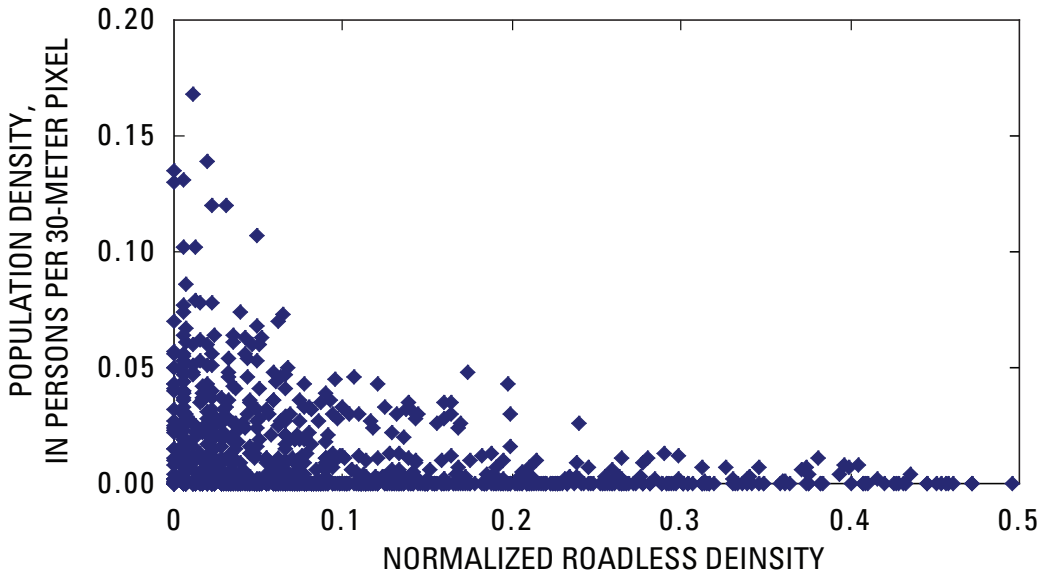

Figure 4. Scatterplot showing the geometrically uneven distribution of data points. The clustering of data near the origin indicates that the highest population densities are found in the areas containing the highest density of roads.

of data points near the origin, the scatterplot shows the data variation in a geometric region of high data concentration. The training data for the experiment above were selected using a simple random-number generator across the range of normalized road density values (from 0 to approximately 0.8). This approach selects more data points from the dense areas of the data cloud and fewer points from the thinner areas. Trying to squeeze out the noise in the fitted curve by increasing the sample size gives more points where they may not be needed for better calibration (the dense portions of the cloud) and perhaps not enough extra points in the areas where they may improve the calibration (the thin portions of the cloud). A better sampling technique would more uniformly distribute the points among the thicker and thinner portions of the data cloud.

Bayesian Samples and Stopping Points

Suppose that after various attempts at efficiency suggest that a large, rather than a small, training sample is actually required for accurate calibration. There may be a stopping point at an appropriate point of diminishing returns, but this point needs to be identified. Because machine-learning makes no assumptions about the functional form of the data or the distribution of error in the data, the convergence criteria must be based on heuristics and judgment. Suppose that rather than taking a calibration sample of 50,000 data points, 10 independent samples of 10,000 data points each are taken. In the above experiment calibration with 10,000 points takes about 1 hour, while calibration with 50,000 points takes about 26 hours. One question is whether selecting five training sets of 10,000 points each (for a total of 5 hours of calibration time)—and properly pooling the outcomes—would give as much information as one set of 50,000 calibration points. Additional questions are whether 26 calibration runs of 10,000 points each give more or less information as one run of 50,000 points, and whether, in 1 to 26 calibration runs of 10,000 points each, there is some way of assessing the improvement at each step to decide a point of diminishing returns has been found. If so, then iteration can be terminated. This would require an overall goodness-of-fit statistic that could be combined with earlier statistics as the runs progress. This is an example of bootstrapping (Effron, 1979). If calibration is formulated as an iterative process, then it might be able to measure diminishing returns using Bayesian sequential statistics (Gelman and others, 2004).

Retraining

Adding more training points to squeeze down noise in calibration data may give a more interconnected network than is actually needed to accurately describe the curve, which may, in turn, give a longer run time. Thus the prediction as well as the calibration time may depend on the geometry of the data cloud: a scattered data cloud—one with a high noise level—leads to more computation than does a compact data cloud—one with a lower level of noise. This suggests that faster run times can be achieved by training the PNN on a small low-noise data set rather than on a larger set that includes a higher level of noise. There is no general way, other than judgment,

of distinguishing noise in a data set from real variation. But once the PNN is trained, the pattern is described as a curve rather than as a data cloud. This suggests that run time can be reduced by retraining. After fitting the curve, the PNN can be used to recalculate the curve at specified grid points. For example, the curves from figure 3 were plotted from fewer than 50 coordinate pairs (x,y). Retraining on the smaller data set gives a similar curve, but a faster run time, than the model produced with the larger training set. In this case, the improvement in run time is not dramatic, but in the multivariate case it might be.

Multivariate Experiments

As in the example of geospatial correlation, it should also be possible to formulate a constrained optimization algorithm to balance cost and benefit. However, it may be that the cost function for a multivariate function is worse than quadratic, for example, a higher order polynomial. The functional form would need to be decided by statistical experiment. For the univariate case the benefit (goodness-of-fit) can be assessed by visualization—plotting the curves to check for convergence. This may not be as straightforward in the multivariate case, and so a different way of assessing goodness-of-fit would have to be considered. There might also be clustering of data points in some regions of the higher dimensional space, so that there needs to be a multivariate version of a sampling algorithm. Once a multivariate function has been fit to the data, retraining could be used as before. If, for example, given a multivariate function of the form $z = z(u, v, w)$ and appropriate partitions of the *u*, *v*, and *w* axes, then retraining of the network could be done as before using the coordinates $(u_i, v_j, w_k; z_{i,j,k})$. The results of retraining could either be applied directly as a neural network or via table look up.

Acknowledgements

A number of people helped make this report possible by providing guidance and data. Peter Pacheco of the University of San Francisco provided access to the University's Keck Cluster, a parallel-processing computer system. From the U.S. Geological Survey, Rachel Sleeter provided dasymetric data for the San Francisco Bay Area and Ray Watts provided the roadless data set for the correlation experiments.

References Cited

Aarts, E., and Korst, J., 1989. Simulated annealing and boltzman machines; a stochastic approach to combinatorial optimization and neural computing: John Wiley & Sons, New York, 272 p.

Albright, B., 2007, An introduction to simulated annealing: The College Mathematics Journal, v. 38, no. 1, p. 37-42.

Bishop, C.M., 2006, Pattern recognition and machine-learning: Springer Series in Information Science and Statistics, 738 p.

Conrads, P.A., Roehl, E. A., Jr., and Martello, W.P., 2002, Estimating point source impacts on the Beaufort River using artificial network models, in Proceedings, The 2002 AWRA Spring Specialty Conference on Coastal Water Resources.

Cucker, F., and Smale, S., 2001, On the mathematical foundations of learning: Bulletin of the American Mathematical Society (new Series) v. 39, p. 1-49. [www.ams.org/publications].

Efron, B.,1979, Bootstrap methods; another look at the jackknife: The Annals of Statistics, v. 7, no. 1, p. 1-26.

Gelman, A., Carlin, J.B., Stern, H.S., and Rubin, D.B., 2004, Bayesian data analysis (2nd ed.): Chapman &Hall/CRC, 668 p.

Hastie, T., Tibshirani, R., and Friedman, J., 2001, The elements of statistical learning; data mining, inference, and prediction: New York, Springer, 533 p.

Masters, T., 1993, Practical neural network recipes in C++: Boston, Academic Press, Inc. 493 p.

Masters, T., 1995, Advanced algorithms for neural networks; a C++ sourcebook: John Wiley and Sons, Inc., 431 p.

Mattson, T.G, Sanders, B.A., and Massingil, B.L., 2004, Patterns for parallel programming: Boston, Addison-Wesley, the software patterns series, 355 p.

Pacheco, P.S., 1997, Parallel programming with MPI: San Francisco, Morgan Kaufmann, 418 p.

Poggio T., and Smale, S., 2003, The mathematics of learning; dealing with data: Notices of the American Mathematical Society, v. 50, no. 5, p. 537-544.

Press, W. H., Teukolsky, S.A., Vetterling, V.T., and Flannery, B.P., 1992, Numerical recipes in C (2nd ed.): Cambridge University Press, 994 p.

Ripley, B.D., 1996, Pattern recognition and neural networks: Cambridge, Cambridge University Press, 403 p.

Risley, J.C., Roehl, E.A., Jr., and Conrads, P.A., 2003, Estimating water temperatures in small streams in western Oregon using neural network models: U.S. Geological Survey Water Resources Investigations Report 2002-4218, 69 p. [http://pubs.er.usgs.gov/pubs/wri/wri024218, accessed February 6, 2006].

Roehl, E.A., Jr., and Conrads, P., 1999, Real-time control for matching wastewater discharges to the assimilative capacity of a complex, tidally affected river basin, in Proceedings of the South Carolina Environmental Conference, Myrtle Beach, March 15-16, 1999, 5 p.

Sleeter, R., 2004, Dasymetric mapping techniques for the San Francisco Bay region, California: Urban and Regional Information Systems Association, Annual Conference, Proceedings, Reno, Nev., November 7-10, 2004.

U.S. Census Bureau, 2007 [http://www.census.gov/, accessed June, 8, 2007].

Watts, R.D., Compton, R.W., McCammon, J.H., Rich, C.L., Wright, S.M., Owens, T., and Ouren, D.S., 2007, Roadless space of the conterminous United States: Science, v. 316, no. 736, p. 736-737.