NN3 Time Series Forecasting with Radial Basis Function Networks

Susan C. White

Abstract—**This describes the methodology used to construct forecasts for the subset of 11 time series in the NN3 competition. All forecasts were calculated in Excel. The author is currently writing code to perform the same steps described herein.**

I. Methodology

Research examining the function approximation properties of neural networks dates back at least to the late 1980s [1]-[9]. It seems natural to use function approximation methods for time series forecasting; however, results have been mixed. This research uses a radial basis function network with multiquadric basis functions to forecast the subset of 11 time series from the NN3 competition.

A radial basis function neural network takes following the basic form:

$$s(x) = \sum_{i=1}^{k} \lambda_i \phi(\|x - C_i\|) \qquad (1)$$

where λ_i represents a coefficient in $\Re$ to be determined, ϕ represents a radial basis function whose form is to be selected, C_i represents a point, or a "center" in $\Re^r$ whose position is to be determined, and $\| \bullet \|$ represents the traditional Euclidean norm; there are k "centers" (k must also be determined), and r is the dimension of the input vector. (See [10]-[13].) This approximation is the weighted sum of functions of norms (which are radially symmetric). This is a "local" approximating function because the distance of each input vector from each of the centers (the C_is) is important in determining the output. The centers will respond differently depending upon their proximity to the input vector. The λ_is are estimated by requiring that *f(x)*, the true value of the function, equal *s(x)* at some (or all) of the points at which *f(x)* has been sampled. Thus, in this case, the problem of estimating the λ_is given a set of points C_i is simply the problem of solving a system of linear equations. The requirement that $f(x) = s(x)$ can be relaxed to $| f(x) - s(x) | \leq \varepsilon$ for some $\varepsilon > 0$ if there is noise present in the data [13].

Radial basis functions (the $\phi(\bullet)$ in equation 2) have typically taken one of the following forms:

$\phi(r) = r$ (linear)
$\phi(r) = r^3$ (cubic)
$\phi(r) = r^2 \log r$ (thin plate spline)
$\phi(r) = e^{-r\hat{}2}$ (Gaussian)
$\phi(r) = \text{sqrt}(r^2 + a^2)$ (multiquadric)
$\phi(r) = 1 / \text{sqrt}(r^2 + a^2)$ (inverse multiquadric)

In the two multiquadric forms, a^2 is a constant which is set by the user. The linear, thin plate spline, and two multiquadric forms satisfy a theory by Micchelli [14] which proves that the matrix in equation 2 will be invertible; thus, the underlying system of equations is guaranteed to have a solution. The output of the RBF network is the weighted sum of the response of each center to the input vector.

Most of the RBF networks used in time series forecasting applications employ the Gaussian radial basis function. However, it is not clear that the Gaussian is the best radial basis function to use for this application. The multiquadric radial basis function will result in an underlying system of equations which will have a solution [14]. Furthermore, it possesses desirable localization properties. Buhmann [15] and Buhmann and Powell [16] used generalized Fourier transforms to examine the decay – or localization – properties of the thin plat spline and the multiquadric forms of RBFs. The result is that, when r (the dimension of the input vector) is even, the cardinal function for the thin plate spline decays to zero exponentially as $\| x \| \rightarrow \infty$. When r is odd, the decay rate is a direct function of the dimension of the problem (as r increases, the decay rate increases). Buhman and Powell [16] report that the decay properties for the multiquadric are similar to those of the thin plate spline when r is odd. The fast decay is a desirable property for function interpolation: a fast decay ensures that "distant" points will have little influence in the interpolation.

Computationally, the thin plate spline and multiquadric radial basis functions provide more information than the Gaussian in that the difference in distances for distant centers in the time domain can be represented meaningfully on a computer. This is not true for the Gaussian. For example, consider two distances, 3 and 9. For the thin plate spline, they would result in values of 9.9 and 178.0 for $\phi \| \bullet \|$. The multiquadric (with the constant in the multiquadric, $a^2 = 35$) would result in values of 6.6 and 10.8. For the Gaussian, they would result in values of 0.000123 and 6.6E-36. The thin plate spline and multiquadric ranges are within a reasonable range for computer calculations; this is not true for the Gaussian. Simulations with varying amounts of noise suggest that the multiquadric RBF is more robust. In addition, Coulomb, et al. [17] also find that the multiquadric is a very robust RBF. Thus, the multiquadric is used for the time series forecasts presented here.

Casdagli [18] suggests the use of radial basis functions to construct a nonlinear mapping of historical time series data; this mapping is then used for prediction. The goal, then, is

Manuscript received May 24, 2007.

S. C. White is with the Department of Decision Sciences, School of Business, The George Washington University, Washington, DC 20052 USA phone: 202-994.4678; fax: 202-994-2736; e-mail: scwhite@gwu.edu

to find an interpolating function of the form of equation 1. This requires determining the number of centers (k), the location of the centers ($\boldsymbol{C}_i$s), and the weights (λ_is). The interpolation approach, so termed by Poggio and Girosi [19], uses every data point in the historical data as a center ($\boldsymbol{C}_i$). This interpolation approach forces the function to pass through each of the historical data points. If there is noise in the data, then an "approximation" approach is preferred to the interpolation approach [13] [19] [20]. In this case, each of the $\boldsymbol{C}_i$s in equation 1 is not required to be a historical data point, and $k \ll N$ where N is the number of observations in the data. In this case, the centers ($\boldsymbol{C}_i$s) are identified first. Then, given a set of centers, the weights (λ_is) are computed via a system of linear equations. Thus, for RBF networks, the forecasting problem reduces to two basic steps: identifying the centers (their number and location) and calculating the λ_is.

There are several basic approaches for locating the centers: k-means clustering (as used by [21]-[22]), a heuristic "resource-allocating" approach [23], or another heuristic-based alternative proposed by Omohundro [24]. Omohundro's approach starts with a center at each of the historic data points and successively merges centers which increase the estimation error the least. The merging is stopped when the estimation error increases by more than some ε, which is problem dependent. Omohundro notes that, in the k-means approach, the centers typically are located where the data is the most dense; in the best-first model merging approach, the centers tend to be located where the function varies the most. Platt's approach [23] allocates a new center when the current set of centers is insufficient to model an input. The model starts with no centers, and a desired accuracy (ε) is set. As inputs are presented to the network, it chooses to store some of them as centers. Platt defines two rules for allocating new centers:

1. An new center should be stored if the input is far away from the existing centers ($\| x - C_{nearest} \| > \xi$, where ξ is problem dependent) and
2. A new center should be stored if the difference between the desired output and the output of the network is too large ($f(x) - s(x) > \varepsilon$).

The combination of the two rules creates a "compact network" ([23], p. 717). Both Omohundro's and Platt's approaches address the problems of network architecture, overfitting and overtraining which are difficult to avoid with traditional neural network models. That is, there is only one hidden layer in the network (see equation 1), and the number of nodes in the hidden layer is determined dynamically. Finally, the "training" issue is simply a matter of inverting a matrix once the centers have been located. Experiments on simulated data suggest that a modification of Platt's approach will work best for time series forecasting. First, the maximum number of centers is set to $N/2$ where N is the number of r-dimensional input vectors available; if $N/2 > 30$, then the maximum number of centers is set at 30 (to avoid overfitting).

The method is coded as follows:

1. The last r-dimensional observation in the training data is selected as the first center.
2. The distance of each subsequent r-dimensional observation in the training data from all existing centers is calculated.

- The current observation is allocated to the closest existing center – unless it is "too far" from any existing center. ("Too far" is defined below.)
- If the current observation is "too far" from an existing center, a new center is allocated. If the new center will result in an acceptable number of centers (i.e., no exceed the maximum), then a new center is allocated at the current observation. If the maximum number of centers will be exceeded by creating the new center, two existing centers are merged OR the "distant observation" is allocated to one of the existing centers. If the distance of the observation to the nearest center is less than the distance between the two closest centers, then it is allocated to the nearest existing center. If the two centers are closer to each other than the current observation is to any existing center, then the two closes centers are merged to form one center, and the current observation is allocated as a new center.
- After every observation has been allocated to a center, any center which has only one observation allocated to it is deleted from the set of centers. (This step is necessary only if there is noise in the data.)

"Too far" is determined heuristically. The data are scaled to [0, 1], so the farthest two points could be apart is $\sqrt{r}$ – the square root of the input dimension. "Too far" is calculated as $\sqrt{r} / \zeta$. For nonseasonal data ζ is 70.0 for annual data; 60.0 for quarterly data; and 50.0 for monthly data. For seasonal data, it is 55.0 for quarterly data and 45.0 for monthly data. (This heuristic borrows the idea that data collected more frequently are noisier from Schnaars and Bavuso, 1986.) The distance for seasonal data is slightly larger because some of what is termed "seasonality" could really be noise. Results presented in the next section suggest that this value should be smaller for short-term forecasts and larger for longer-term forecasts. (This is consistent with Lowe and Webb's finding (1991) that networks should be trained to more rigorous tolerances for short-term forecasts. The smaller distance results in a network which mimics the historical data more closely.)

One final parameter must be determined: the number of inputs to the model, r, or the "dimension" of the problem. This was determined using correlations: up to a maximum of the five strongest lags were and a heuristic test for a trend (which examined runs of increases or decreases) was used to detect data with a trend. For data with a trend, the first differences were used in the RBF model. (Lohninger [25] indicates that the RBF approach will perform poorly for data with a linear trend.)

The centers were then located as described above and the forecasts were calculated. k-step forecasts were calculated iteratively (with previous forecasts as inputs). If the forecasts "blew up" (became dramatically large), then ζ was

incremented by 5 and the process repeated. Finally, for detrended data, the sums are calculated for the forecasts and seasonality is put back into the data.

REFERENCES

[1] G. Cybenko, "Continuous Valued Neural Networks: Approximation Theoretic Results," *Computer Science and Statistics: proceedings of the 20th Symposium on the Interface*, 174 – 183, 1988.

[2] K. Hornick, M. Stinchcombe and H. White, "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, 2, 359 – 366, 1989.

[3] M. Stinchcombe and H. White, "Universal Approximation Using Feedforward Networks with Non-Sigmoid Hidden Layer Activation Functions," *Proceedings of the International Joint Conference on Neural Networks*, I – 613 – 617, 1989.

[4] M. Stinchcombe and H. White, "Approximation and Learning Unknown Mappings Using Multilayer Feedforward Networks with Bounded Weights," *Proceedings of the International Joint Conference on Neural Networks*, III – 7 – 16, 1990.

[5] E. J. Hartman, J. D. Keeler, and J. M. Kowalski, "Layered Neural Networks with Gaussian Hidden Units as Universal Approximations," *Neural Computation*, 2, 210 – 215, 1990.

[6] J. Park and I. W. Sandberg, "Universal Approximation Using Radial-Basis-Function Networks," *Neural Computation*, 3, 246 – 257, 1991.

[7] E. K. Blum and L. K. Li, "Approximation Theory and Feedforward Networks," *Neural Networks*, 4, 511 – 515, 1991.

[8] W. Light, "Ridge Functions, Sigmoidal Functions, and Neural Networks," in E. W. Cheney, C. K. Chui and L. L. Schumaker (Eds.), *Approximation Theory VII*, Academic Press, Boston, 163 – 206, 1992.

[9] B. Lenze, "Constructive Multivariate Approximation with Sigmoidal Functions and Applications to Neural Networks," in D. Braess and L. L. Schumaker (Eds.), *Numerical Methods of Approximation Theory, Vol. 9*, Birkhäuser Verlag, Basel, 155 – 175, 1992.

[10] M. J. D. Powell, "Radial Basis Functions for Multivariable Interpolation: A Review," in J. C. Mason and M. G. Cox (Eds.), *Algorithms for Approximation*, Clarendon Press, Oxford, 143 – 167, 1987.

[11] M. J. D. Powell, "The Theory of Radial Basis Function Approximation in 1990," in W. Light (Ed.), *Advances in Numerical Analysis, Vol. II*, Clarendon Press, Oxford, 105 – 210, 1992.

[12] D. S. Broomhead and D. Lowe, "Multivariable Function Interpolation and Adaptive Networks," *Complex Systems*, 2, 321 – 355, 1988.

[13] M. Casdagli, "Nonlinear Prediction of Chaotic Time Series," *Physica D*, 35, 335 – 356, 1989.

[14] C. A. Micchelli, "Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions," *Constructive Approximation*, 2, 11 – 22, 1986.

[15] M. D. Buhmann, "On Quasi-Interpolation with Radial Basis Functions," *Journal of Approximation Theory*, 72, 225 – 255, 1990.

[16] M. D. Buhmann and M. J. D. Powell, "Radial Basis Function Interpolation on an Infinite Regular Grid," in J. C. Mason and M. G. Cox (Eds.), *Algorithms for Aproximation II*, Chapman & Hall, London, 146 – 169, 1989.

[17] Coulomb, J.-L., A. Kobetski, M.C. Costa, Y. Marechal, and U. Jonsson, 2003, "Comparison of radial basis function approximation techniques," *Compel*, 22, 616 – 629.

[18] M. Casdagli, "Nonlinear Prediction of Chaotic Time Series," *Physica D*, 35, 335 – 356, 1989.

[19] T. Poggio and F. Girosi, "A Theory of Networks for Approximation and Learning," A. I. Memo No. 1140, MIT Artificial Intelligence Laboratory, 1989.

[20] C. Bishop, "Improving the Generalization Properties of Radial Basis Function Neural Networks," *Neural Computation*, 3, 579 – 588, 1991.

[21] J. Moody and C. Darken, "Learning with Localized Receptive Fields," in D. Touretzky, G. Hinton, and T. Sejnowski (Eds.), *Proceedings of the 1988 Connectionist Models Summer School*, Morgan Kaufmann, San Mateo, CA, 133 – 143, 1989.

[22] D. Wettschereck and T. Dietterich, "Improving the Performance of Radial Basis Function Networks by Learning Center Locations," in J. E. Moody, S. J. Hanson, and R. P. Lippman (Eds.), *Advances in Neural Information Processing Systems 4*, Morgan Kauffman, San Mateo, CA, 1133 – 1140, 1992.

[23] J. C. Platt, "Learning by Combining Memorization and Gradient Descent," in R. P. Lippman, J. E. Moody, and D. S. Touretzky (Eds.), *Advances in Neural Information Processing Systems 3*, Morgan Kauffman, San Mateo, CA, 714 – 720, 1991.

[24] S. M. Omohundro, "Best-First Model Merging for Dynamic Learning and Recognition," in J. E. Moody, S. J. Hanson, and R. P. Lippman (Eds.), *Advances in Neural Information Processing Systems 4,* Morgan Kauffman, San Mateo, CA, 958 – 965, 1992.

[25] H. Lohninger, "Evaluation of Neural Networks Based on Radial Basis Functions and Their Application to the Prediction of Boiling Points from Structural Parameters," *Journal of Chemical Information and Computer Sciences*, 33, 736 – 744, 1993.