Presented at the 2009 ISPA/SCEA Joint Annual Conference and Training Workshop - www.iceaaonline.com

Parametric Cost Analysis Using Neural Networks

by
Edwin B. Dean
Consultant
designforvalue@att.net

Neural Networks: How They Work

- A neural network is a mathematical entity that simulates the learning capability of the brain. It learns by approximating a set of outputs given a set of inputs.
- It has multiple inputs and multiple outputs. The result of the learning process is a model that predicts the desired outputs based upon a set of (input, output) data.
- A neural network has an input layer, one or more hidden layers, and an output layer.
- The input layer contains all of the inputs. Each hidden layer and the output layer contain one or more artificial neurons.
- Each artificial neuron in the output layer corresponds to a particular output.
- Each artificial neuron has multiple inputs, a threshold, an activation function, and a single output. The activation function is a nonlinear function, such as the inverse tangent or the logistic function, that provides the learning capability.

Neural Networks: How They Learn

- The learning occurs by applying the (inputs, outputs) data to the network. As each set of input data is applied, the weight of each input to and the level of the threshold for each artificial neuron is adjusted to provide a better prediction of the set of outputs.
- A single application of all of the (input, output) data is called an epoch. Epochs are applied until the inputs provide a reasonable approximation of the outputs.
- There are many types of neural networks. The only type addressed in this paper is the backpropagation model originally developed by Werbos (1974) and, later, independently developed by Rumelhart, Hinton, and Williams (1986).
- The backpropagation network attempts to reduce the root mean square of the error between the outputs of the neural network output layer and the output data.

Prior Application of Neural Networks for Cost Analysis

- The primary publications of neural networks applications cost analysis have occurred within the Association for the Advancement of Cost Engineering International (AACEI) community
- The first neural network for cost analysis published papers were McKim (1993a) and McKim (1993b)
    - Both McKim (1993a) and McKim (1993b) introduce the concept of the backpropagation neural network and provide an excellent introduction to the mathematics involved
    - McKim (1993b) reports that "The neural network trained for this paper was able to predict the cost overruns of ten projects with approximately half the error of more conventional statistical methods"
- Today, when one searches on the AACEI library using the keyword "neural network", twenty abstracts appear
- Early papers include the following:
    - Creese and Li (1995) estimate the cost of timber bridges using a neural network
    - Garza and Rouhana (1995) describe the mathematics of the backpropagation neural network and then compare the results of a neural network prediction with regression predictions
    - Rowings and Sonmez (1996) apply both regression and neural networks for the analysis of labor productivity
    - Moselhi and Siqueira (1998) use a neural network to estimate the cost of structural steel buildings
    - Al-Tabtabai, Alex, and Tantash (1999) propose applying subjective data to a neural network to predict preliminary highway construction costs

Contrived Data for a Cost Model

▪ The following neural network cost model is based upon contrived data because

- o We want to see how well both regression and a neural network fit a known equation
- o We want to see how closely the parameters ***a*** and ***b*** estimated by the regression deviate from the actual parameters a and b

▪ A typical parametric cost model is of the form $y=ax^b$ where
  - o y is the cost
  - o x is weight
  - o a is the complexity and
  - o b is rate of change of cost with respect to weight.

▪ If we were developing a regression model, we would transform the data and use the equation ln(y) = c + b ln(x), where c = ln(a), to estimate b and c.

▪ The data for this example is derived by choosing b and c and then selecting b_i and c_i for the data from a normal distribution centered on b and c for a given x_i to obtain y_i.

▪ At right are the data where weight x is used to estimate the simulated cost y,. The data z are $z = ax^b$ prior to noise being added.

▪ $\{(x_i, z_i)\}$ is the set of points of the equation underlying the simulation.

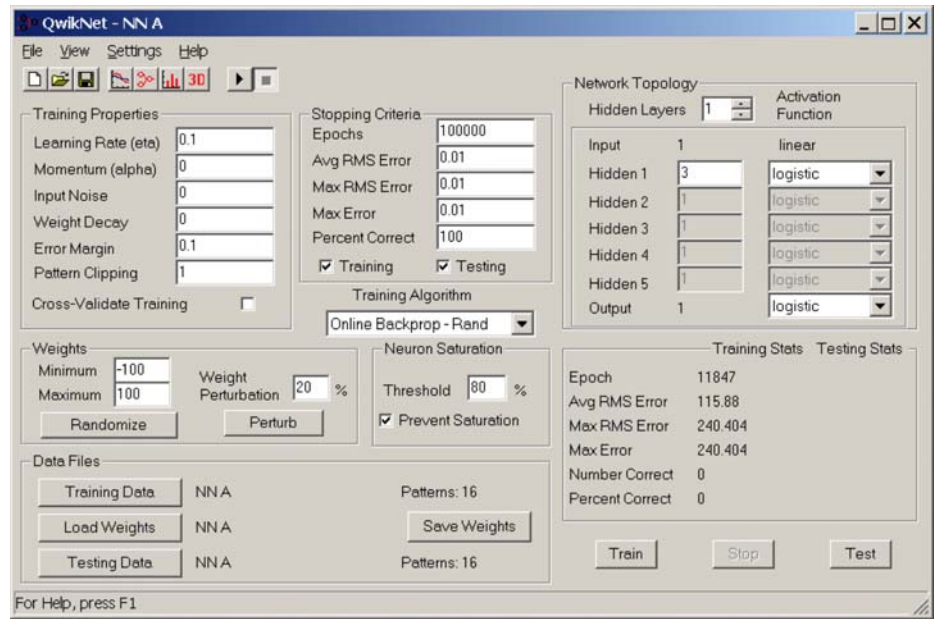
x	y	z
100	386.56565	185.604698
108.044801	263.437665	195.934844
137.605224	135.236046	232.077698
177.879093	339.800484	277.764282
218.933064	273.434789	321.223206
218.94707	432.496933	321.237591
255.354506	384.975446	357.758064
280.076661	302.904774	381.665415
295.537837	592.443618	396.294575
308.377906	344.494947	408.269787
343.312423	540.576817	440.12053
398.480426	410.729949	488.51045
450.550024	399.290353	532.364486
497.332603	859.150582	570.481828
551.212162	574.435695	613.072898
715.503408	827.97908	735.897214

Since the backpropagation neural network can approximate nonlinear equations, we do not have to transform the data, although we could if we wished to do so.

The Neural Network Software Setup

The shareware QuikNet (Jensen, C. (2003)) neural network software is used

- As can be seen from the control panel, the neural network has
  - o 1 input, weight
  - o 1 output, cost
  - o 1 hidden layer with
    - •3 neurons.
- A logistic function is used as the activation function.
- Backpropagation with random perturbation is used as the training algorithm.
- The same data is used for training and testing.

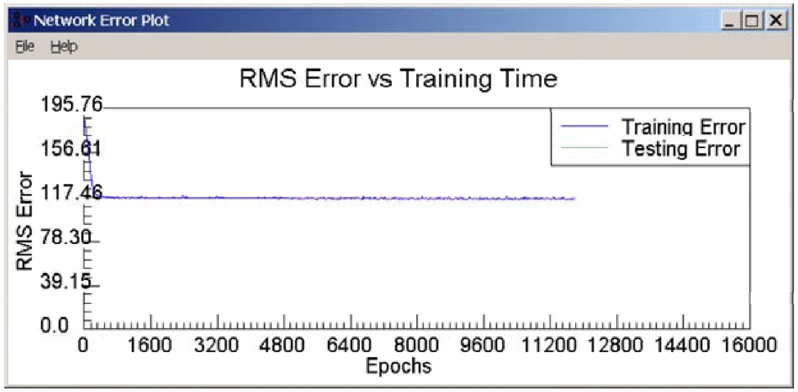


- The training process was allowed to run for 11,847 epochs.
- Each epoch is the application of each of the data points

Training The Neural Network

- Root Mean Square (RMS) Error (RMSE) is the goodness of fit measure
- In the figure below
    - o the RMSE declines rapidly
    - o then reaches an almost steady state level at which the training is assumed to be complete

- The backpropagation model uses nonlinear optimization
    - o Solutions can be caught at sub-optimal points
- The author typically has not encountered suboptimal solutions with backpropagation when
    - o using one or two hidden layers with
        - • one or two inputs
        - • one or two outputs
- Problems can occur for more complex models with
    - o A number of subjective data inputs
    - o Two or more hidden layers



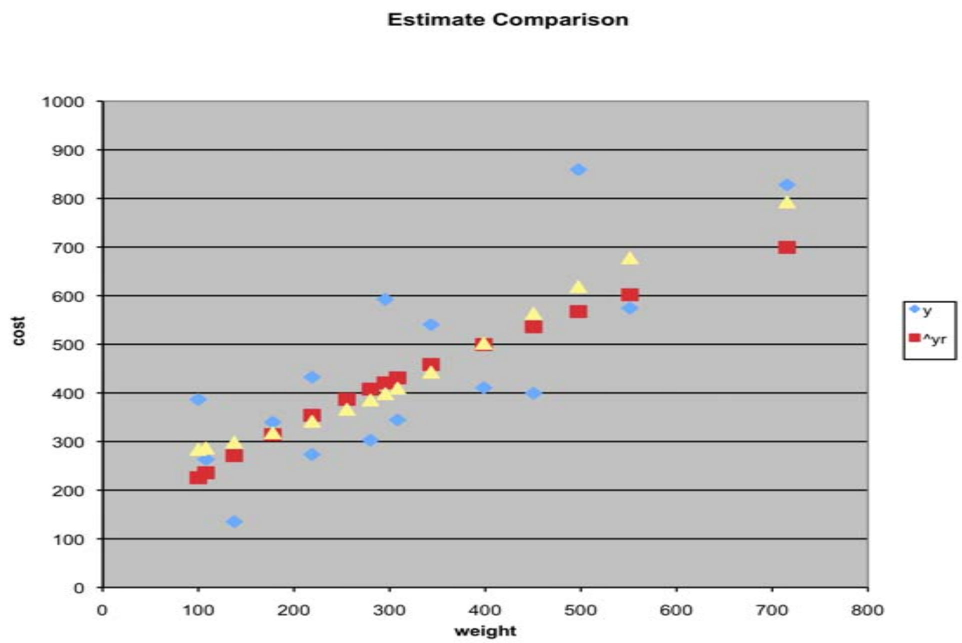
- The larger and more subjective the set of inputs and outputs, the greater the art required in training.

Analyzing Neural Network Model Results

- When we analyze the results of a regression model of the data, we rely on the well-established methods of statistics
- The equations of statistics for regression do not apply to neural networks
- The goodness of fit
  - For regression is R^2
  - For neural networks is Root Mean Square Error (RMSE)
- The author uses two approaches to provide a comparison of regression and neural network results
  - Visual analysis
  - The Angle

Visual Analysis

- To view neural network results we just plot a graph of
  - o The data and
  - o The predicted data points
- To compare neural network results to regression results, we just add the regression predicted data points to the same graph
  - o The cost data points are diamonds
  - o The regression line data points are squares
  - o The neural network predicted data points are triangles
- 3 dimensional charts can display cost surfaces over parameter pair planes



- The neural network predicted curve is nonlinear
- The regression predicted curve follows the curve specified by the equation used for the fit

Presented at the 2009 ISPA/SCEA Joint Annual Conference and Training Workshop - www.iceaaonline.com

Comparing With The Angle

The mathematical basis of the angle is described in Dean (2008). Consider the points x_i y_i $\hat{yr}_i$ and ynn_i to be dimensions within the vectors $\vec{x}$, $\vec{y}$, $\hat{\vec{y}}_r$ and $\hat{\vec{y}}_{nn}$, respectively.

The angle in degrees between vectors $\vec{u}$ $\vec{v}$ is

$$angle(\vec{u},\vec{v}) = \frac{180}{pi}\arccos\left(\frac{\vec{u}\bullet\vec{v}}{\sqrt{\vec{u}\bullet\vec{u}}\sqrt{\vec{v}\bullet\vec{v}}}\right)$$

Applying this formula to the results of our neural network and regression models on the data describe above we have

$$angle(\vec{y},\vec{y}_{nn}) = 13.94 \qquad angle(\vec{y},\vec{y}_r) = 14.56$$

- The fitted intercept coefficient is 2.7717 vs. 2.0000 for the noiseless equation.
- The fitted power is 0.5749 vs. 0.7000 for the noiseless equation

What and Why The Angle?

When we apply regression, we use R^2 as our goodness of fit. Regression uses mean adjusted data for its calculations. Wickens (1995) demonstrates that R, the multiple correlation coefficient, is the cosine of the angle between the mean adjusted data vector and the predicted mean adjusted data vector.

We can thus think of the cosine of the angle we just used as the multiple correlation coefficient of the non mean adjusted data.

Using the angle rather than the cosine of the angle as our goodness of fit measure provides a powerful and intuitive geometric meaning to goodness of fit.

When we apply regression on nonlinearly transformed data, R and R^2 are measures on the space of the transformed data and cannot be projected onto the non transformed data space. However, we really want to know how well the fit is on the non transformed data space. The angle does that for us.

Because the angle works with independent spatial dimensions, the angle is a valid measure of the goodness of fit of predicted data to data for any means of prediction.

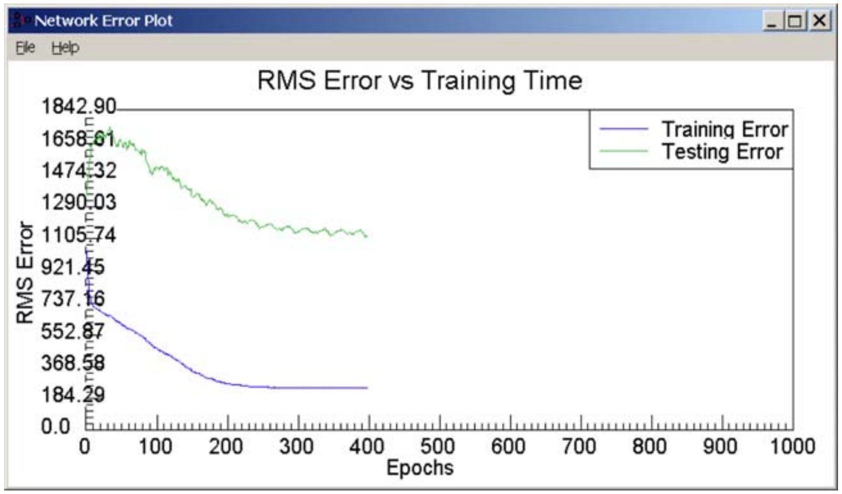
An Instructive Example

- The data comes from the Encyclopedia Astronautica Launch Vehicle Summary Data (Wade (1997-2008))
- All launch vehicles that do not have the desired cost variables and performance parameters are eliminated
- A 5 input, 1 hidden layer with 8 neurons, 2 output neural network (1.8.2) is used to model
    - The development cost and
    - The recurring production cost
- Using performance parameters
    - Total length
    - Core diameter
    - Span
    - Total mass and
    - Maximum range
- A logistic activation function is used
- A randomly perturbed backpropagation training algorithm is applied to the data
- The results are compared to the results of a 5 parameter multivariate regression for each cost using the same performance parameters
- Graphing the costs against the performance parameters indicates that log transformed data should be used for the regressions
- The neural network inputs are the untransformed performance parameters

The Training and The Results

- The data has been split into a training data set and a test data set
- The neural network is trained on the training data set
- Test data set is used
  - o To monitor the training process and
  - o To test how well the neural network generalizes to similar but different data

- The error reduction of the training process at right shows the stabilization of the error of the test set above and the training set below.
- If allowed to train further the error of the test set rises as the network over trains on the training data at the expense of the error over the test set.



And the results are >
The smaller, the better

Goodness of Fit (degrees)	Multivariate Regression Training Data	Neural Network Training Data	Multivariate Regression Test Data	Neural Network Test Data
Development Cost	31.6	14.2	72.4	8.40
Recurring Production Cost	11.7	14.9	89.8	34.5

Getting Started with Neural Networks

- The AACEI references in this paper provide adequate information to understand the mathematics underlying the backpropagation neural network
    - o They may be obtained at the AACEI website (http://www.aacei.org/)
        - Search for them at the bookstore using the keyword “neural network”
        - AACEI members can download the papers for free

- Jones and Hoskins (1987) and Hecht-Nielsen (1992) do an excellent job of explaining the inner workings of the backpropagation neural network
- For a detailed introduction to several of the more popular types of neural networks, including backpropagation, study Lippmann (1987).
- Haykin (1994, 1998) is the comprehensive source for learning about all types of neural networks
- Trippi and Turban, ed. (1996) provide a variety of applications to finance in general

- By downloading the shareware program QuikNet (Jensen (2003)), one can begin to get the feel of training a neural network and actually develop models
    - o The author uses this software under Windows Professional 2000 Pro
    - o As most software does, QuikNet has a few quirky bugs
    - o Hopefully it will work under Windows XP and Windows Vista

- For more sources for learning about neural networks or for more advanced neural network software, perform a web search for “neural network” or “neural network software”

Summary and Conclusion

- There is a learned art to the development of neural network models
- Neural networks
    - o Can be used to develop project measure models, including cost and schedule as outputs from a set of input parameters, some of which could be subjective in nature
    - o Facilitate multi-input, multi-output model development
    - o Are inherently nonlinear which opens many modeling possibilities restricted by linear models and related data transformation assumptions
    - o Are not based upon statistics, which requires a substantial number of data points for validity
        - There is no validity restriction on how few data points can be used to develop the model
- The use of the angle between data and predicted data vectors
    - o Provides an intuitive and understandable goodness of fit measure on the space of the non transformed data
    - o Permits comparison of the results of all types of models

- Finally, the experience of the author is that properly trained neural network models usually provide a closer match to the data than do equivalent regression models
    - o This results from the inherent nonlinearity of the models

References 1

AACEI. Association for the Advancement of Cost Engineering International, http://www.aacei.org/.

Al-Tabtabai, H., A. Alex, and M. Tantash (1999). “Preliminary Cost Estimation of Highway Construction Using Neural Networks”, *Cost Engineering*, Vol. 41, No. 03 AACE International, Morgantown, WV.

Creese, R. C. and L. Li (1995). “Cost Estimation of Timber Bridges Using Neural Networks”, *Cost Engineering*, Vol. 37, No. 05, AACE International, Morgantown, WV.

Dean, E. B. (2008). “The Angle: A Goodness of Fit Metric for Cost Analysis”, Proceedings of the ISPA/SCEA 2008 Joint International Conference, 12-16 May 2008, Noordwijk, The Netherlands.

Garza, J. de la and K. G. Rouhana (1995). “Neural Networks Versus Parameter-Based Applications in Cost Engineering”*, Cost Engineering*, Vol. 37, No. 02, AACE International, Morgantown, WV.

Haykin, S. (1994, 1998). Neural Networks: A Comprehensive Foundation, Macmillan College Publishing Company, New York, NY.

Hecht-Nielsen, R. (1992). “Theory of the Backpropagation Neural Network”, in Neural Networks for Perception, Vol. 2, Harcourt Brace & Co., Orlando, FL. pp. 65-93.

Jensen, C. (2003). QuikNet v2.23, http://qwiknet.home.comcast.net/~qwiknet/.

Jones, W. P. and J. Hoskins (1987). “Back-Propagation: A Generalized Delta Learning Rule”, *Byte*, October, pp. 155-162.

References 2

Lippmann, R. P. (1987). "An Introduction to Computing with Neural Nets", *IEEE ASSP Magazine*, pp. 4-22.

McKim, R. A. (1993a). "Neural Network Applications to Cost Engineering", *Cost Engineering*, Vol. 35, No. 07, AACE International, Morgantown, WV.

McKim, R. A. (1993b). "Neural Networks and Identification and Estimation of Risk", 1993 AACE International Transactions, P.05, AACE International, Morgantown, WV.

Moselhi, O. and I. Siqueira (1998). "Neural Networks for Cost Estimating of Structural Steel Buildings", 1998 AAEE International Transactions, IT/IM.06, AACE International, Morgantown, WV.

Rowings, J. E. Jr. and R. Sonmez (1996). "Labor Productivity Modeling with Neural Networks", 1996 AACE International Transactions, PROD.01, AACE International, Morgantown, WV.

Rumelhart, D. E, G. E. Hinton, and R. J. Williams (1986), "Learning Representations by Back-Propagating Errors", *Nature*, 323, pp. 533-536.

Trippi, R. R. and E. Turban, ed. (1996). Neural Networks in Finance and Investing, McGraw-Hill, New York, NY.

Wade, M. (1997-2008). Launch Vehicle Summary Data, Encyclopedia Astronautica, http://astronautix.com/data/lvs.xls .

Werbos, P, J. (1974). Beyond Regression: New Tools for Prediction and Analysis in The Behavioral Sciences, PhD Dissertation, Harvard University, Cambridge, MA.

Wickens, T. (1995). The Geometry of Multivariate Statistics, Lawrence Erlbaum Associates, Hillsdale, NJ, USA.