Estimating Neural Network Nonparametric Production Functions
Using SAS® Software

Daric Brummett, Department of Economics, University of Notre Dame
Lawrence Marsh, Department of Economics, University of Notre Dame

Introduction
The argument of this paper is that neural networks provide a viable basis for nonparametric modeling of production functions. The contribution of this project is to operationalize a nonparametric neural network production model with known statistical properties using SAS/IML® software and the NLIN procedure.

Flexibility of Neural Network Models
Neural network models are weighted linear combinations of nonlinear functions (in this case, logistic functions) which use linear combinations of observed values as their inputs. The models are of the sort

\[ F(X) = \sum_{i=1}^{k} \alpha_i \left[ (1 + \exp(-X\beta_i))^{-1} \right] \]

If obs is the number of observations, n is the number of inputs to the production process, and k is the number of sigmoids in the neural network, then \( \beta_i \) is an \( n+1 \) dimensional column vector of coefficients to be estimated, \( X = [-1, X] \) is an \( \text{obs} \times (n+1) \) dimensional matrix in which -1 is an \( \text{obs} \) dimensional column vector and \( X \) is an \( \text{obs} \times n \) dimensional matrix of input values. Using this notation, a \( k \) dimensional column vector, \( \alpha \), and an \( (n+1 \times k) \) dimensional matrix, \( \beta \), may be created. \( \alpha \) and \( \beta \) are the weight matrices that define the network. As \( k \) increases one unit in magnitude, the number of parameters increase by \( n+2 \) (n additional beta coefficients associated with the inputs, one beta coefficient associated with an intercept, and one alpha coefficient associated with the additional sigmoid). In this way, \( k \) serves as an index to the depth of parameterization.

The flexibility of these models has been documented by White [1992], who built on the work of Kolmogorov [1957], Hecht-Nielsen [1990a], and Cybenko [1988] to show that neural networks are Sobolev flexible and thus qualify as universal approximators. Universal approximators are able to asymptotically approximate any function and all its derivatives to any degree desired. Since neural networks are capable of universal approximation, they are suitable tools for nonparametric analysis of unknown mapping functions.1

Additionally, White [1989b, 1991] has shown the asymptotic consistency of estimates gained utilizing a gradient descent algorithm known as backpropagation in conjunction with Newton’s method.

The flexibility of neural network models is a necessary but not a sufficient condition for nonparametric modeling of finite sample data. A method to determine optimal depth of parameterization is needed to solve the bias/variance problem associated with model generalization. If there are too few parameters (\( k \) is too small), the model will fail to capture all the structure in the data, and high bias will result. If there are too many parameters (\( k \) is too large), the model will fit noise as well as structure, and high variance will result. Simply assuming some arbitrary value for \( k \) to be appropriate and minimizing error will not serve as a viable nonparametric model.

Fortunately, White [1989a, 1991] has suggested a test to determine the optimal depth of parameterization for a given finite sample. The test is a conditional moment test and is in part based on the work of Bierens [1990]. The test is conducted on an estimated network of a given size. The null hypothesis is that the model is correctly specified; the alternative hypothesis is that the model requires an increase in the depth of parameterization. The test randomly searches over a specified region of weight space to determine if the errors from the fitted network are correlated with randomly weighted sigmoid functions called ‘phantom units’. If so, the null hypothesis is rejected.

Starting with \( k \) set to a small number (say, 1) and iteratively applying White’s test as \( k \) is increased until the null hypothesis is not rejected provides a method for avoiding under or over fitting.

Thus, neural network models have been shown to be flexible as well as compatible with a procedure to bypass the bias/variance problem with finite sample testing. These two qualities indicate neural network models are appropriate for nonparametric analysis.

The next section will discuss the estimation of neural network models as well as the implementation of White’s test for parameter depth.

Implementation Using SAS/IML Software
Neural network models are subject to nonlinear optimization techniques to estimate \( \alpha \) and \( \beta \). White [1989b,1991] has suggested that gradient descent and Newton’s method will asymptotically provide consistent estimates. This may be achieved using PROC NLIN and METHOD=GRADIENT for a first round of approximation followed by a round of METHOD=NEWTON using the GRADIENT estimates as starting values for the NEWTON round.

Along with White’s theoretical suggestions for consistency, he admits the actual practice of estimating highly nonlinear equations is somewhat less testing. Asymptotically, with infinite computers, samples and time, White’s asymptotic consistency is of considerable import. However, in practice estimation takes place in the finite world. Multistart algorithms are often utilized and when a certain level of ‘frustration’ is reached, the best estimate is

1Joerding [1991] has considered the restriction of neural network production models to the neoclassical function space, thus making neural network models suitable for flexible form modeling and parametric modeling analysis.
assumed to be the global minimum. So, whatever method provides the smallest error before the level of frustration is reached, is preferable.

This is not necessarily the case with all nonparametric procedures, where a tighter fit may imply over-fitting. However, in this case White’s test will provide an indication of the proper number of parameters. White’s test will restrict the number of parameters in order to prevent over-fitting. Thus, the smallest error for a given number of parameters is always preferable.

In this project, a method has shown itself to be quite successful. First, the parameters are set to small initial random values and METHOD=NEWTON is used to solve the equation in PROC NUN. The parameter estimates are then used to start a second round with METHOD=MARQUARDT. Finally, the fitted estimates are used to start a third round where METHOD=GRADIENT. This three round technique seemed to provide better estimates more quickly than White’s suggestion.

Now that the network is estimated for a given number of parameters, White’s test must be performed to determine if more parameters are necessary. The following SAS/IML code will operationalize White’s test.

```
START HTEST;
    * FIRST, FIND DERIVATIVES OF THE ACTIVATION FUNCTIONS;
    DERIW = Z;  *GET DERIVATIVES OF PARAMETERS;
    DERIVBZ = { (J(OBS,H,1)-Z')*(J(OBS,1,1))' };
    DO COUNT .. I TO OBS;
        TEMP1 .. DERIVBZ(ICOUNT,I);
        TEMP2 .. TEMP1'*(X(ICOUNT,I));
        TEMP3 .. SHAPE(TEMP2,1);  *DEFINE AS SINGLE VECTOR;
        IF COUNT = 1 THEN DERIW = TEMP3; ELSE DERIW = DERIW + TEMP3;
    END;
    DERIVMAT = DERIW || DERIW || DERIVBZ;  *MATRIX OF DERIVATIVES;

    * SECOND, CALCULATE WHITES STATISTIC;
    STARTER = 0;
    FAILCOUNT = 0;
    SSE = 0;
    GAMMA = 0;  *START THE RANDOM PROCEESS;
    N=NCOL(X);
    DO I = 1 TO 10 BY 1;
        RAN = INT( GAMMA[NROW(GAMMA),1] ) + 1;  * RANDOM SEED FROM GAMMA;
        DO J = 1 TO I BY 1;
            P = I;  * CHECKING 10 PHANTOM H UNITS;
            PSTAR = J;  * WILL USE 3 PRIN COMP;
            TOLER = 9;  * REJECTION REGION SPECIFIED;
            GAMMA = J(N+1,P,RAN);  * GENERATE RANDOM VECTORS;
            GAMMA = RANDUN(GAMMA);  *SCALE RANDM CORRECTLY;
            PHII = (1+EXP(-NET))#-1;
            CALL SVD(U,D,T,PHII);
            D = DIAG(D);
            PHITEST = PHII;
            PHISTAR = PHISTAR*(1;PSTARI);
            RUN REGRESSION
            ONES = J(OBS,1,1);
            XTEST = ONES || DERIVMAT || PHISTAR;
            YTEST = E;
            BTTEST = GINV(XTEST)*XTEST*YTEST;
            YTESTHAT = YTESTHAT*YTEST;
            ETEST = YTEST - YTESTHAT;
            TSS = TE*TE;
            SSE = ETEST*ETEST;
            R2 = 1'SSE/TSS;
            STAT = OBS*R2;
            IF STAT > OBS*PSTAR THEN LABEL = 'FAIL'; ELSE LABEL = 'PASS';
            IF LABEL = 'FAIL' THEN FAILCOUNT = FAILCOUNT + 1;
            IF STARTER = 0 THEN OUTPUT = TEMP; ELSE OUTPUT = OUTPUT + TEMP;
            IF STARTER = 0 THEN OUTLAB = LABEL; ELSE OUTLAB = OUTLAB||LABEL;
            STARTER = 1;
        END;
    END;
    NAMEPROB = {'P' 'PSTAR' 'SSE' 'STAT' 'PROB' 'LABEL'};
    POSSIBLE = NROW(OUTPUT);
    PERCENT = FAILCOUNT/POSSIBLE;
    RESULTS = FAILCOUNT // POSSIBLE // PERCENT;
    RN = {'#FAILS' 'TRIALS' 'PERCENT'};
    PRINT OUTPUT((COLNAME=NAMEPROB)) OUTLAB RESULTS((ROWNAME=RN));
    FINISH HTEST;
```

2 For further explanation and theoretical justification of this code, see White [1989b, 1991, 1992].

3 Mathematically, Z = [Z1, Z2, ..., Zk] where

```
Zi=\begin{bmatrix} 1+\exp(-X_{i1}) \\ \vdots \\ 1+\exp(-X_{ik}) \end{bmatrix}^{1/3};
```

Where β = [BIASZ,W], and α = V and Z is an (obs x k) matrix of output values for each sigmoid in the model (corresponding to the columns of Z) and for each element of observed data (corresponding to the rows of Z). The result of running the module HTEST is a vector of trials. Ultimately the goal is to have no rejections of the null hypothesis. If no rejections occur, the output PERCENT will be 0. Any other value for PERCENT indicates further parameters may be necessary.

Starting with a low number of parameters (say k=1) will produce a small network. The network may easily be estimated by PROC NLIN and then tested using the module HTEST. If rejection of the null hypothesis occurs, an increases in the depth of parameterization is appropriate (say to k=2). Then the network parameters could be estimated by PROC NLIN and the model tested by HTEST.
Iteratively applying this algorithm will produce an 'optimal' network model.

Evaluation of Network Performance

Network performance against parametric models of production could be determined using non-nested likelihood ratio testing. Each model would be tested against all other models. In each test, the model with more parameters would be considered the unrestricted model, while the model with fewer parameters would be considered the restricted model. A likelihood ratio test could be performed to determine the performance of the network model relative to the restricted model. In this way the appropriateness of the unrestricted model could be evaluated as compared to other models.

Results

A data set of coal production in the US with 245 observations was used. Each observation had four inputs (three capital variables and one labor) and output measured in raw metric tons of coal produced. Neural network models with increasing depths of parameterization were estimated and tested. Depth of parameterization started at $k=1$ and progressed until the null hypothesis of White's test was not rejected. The following table indicates results.

<table>
<thead>
<tr>
<th>MODE</th>
<th>PERCENT</th>
<th>R2</th>
<th>ADJR2</th>
<th>PARAMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN1</td>
<td>1</td>
<td>.733</td>
<td>.727</td>
<td>6</td>
</tr>
<tr>
<td>NN2</td>
<td>2</td>
<td>.782</td>
<td>.771</td>
<td>12</td>
</tr>
<tr>
<td>NN3</td>
<td>3</td>
<td>.865</td>
<td>.854</td>
<td>18</td>
</tr>
<tr>
<td>NN4</td>
<td>4</td>
<td>.899</td>
<td>.878</td>
<td>24</td>
</tr>
</tbody>
</table>

At a parameterization of $k=4$, with 24 parameters, the model was assumed optimal because PERCENT of trials rejecting the null hypothesis of White's test dropped to zero. Coefficients of determination are also shown (R2 and adjusted R2) which rise with the degree of parameterization, indicating an increasing fit.

Alternative models were also estimated including parametric and flexible form models. The parametric models included Linear (LIN), Quadratic (QUAD), Cobb-Douglas (CD), CES (CES), Transcendental (TRANS), and Transcendental Logarithmic (TLOG). The flexible form models were the Box-Cox models with only the dependent variable transformed (BC0), the dependent and independent variables transformed by a single $\lambda$ (BC1), and the dependent and independent variables being transformed by two $\lambda$'s (BC2). The following table shows results.

<table>
<thead>
<tr>
<th>MODEL</th>
<th>R2</th>
<th>ADJR2</th>
<th>PARAMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIN</td>
<td>.772</td>
<td>.768</td>
<td>5</td>
</tr>
<tr>
<td>QUAD</td>
<td>.824</td>
<td>.813</td>
<td>15</td>
</tr>
<tr>
<td>CES</td>
<td>.843</td>
<td>.839</td>
<td>8</td>
</tr>
<tr>
<td>CD</td>
<td>.857</td>
<td>.855</td>
<td>5</td>
</tr>
<tr>
<td>TRANS</td>
<td>.873</td>
<td>.869</td>
<td>9</td>
</tr>
<tr>
<td>TLOG</td>
<td>.876</td>
<td>.869</td>
<td>15</td>
</tr>
</tbody>
</table>

The likelihood ratio testing of non-nested models requires a transformation of the error of the two models such that their units are the same. While it is true that errors often may not be transformed prior to estimation without violation of the assumptions of estimation, errors may be transformed after estimation with no such danger.

The transformation is as follows. Each model has an R2 which is unit free. The original data generated a total sum of squares (TSSO) which is unit dependent. Using the R2 measures from each model, a sum of squared errors (SSEO) measurement for each model may be obtained which has the units of the original data regardless of what transformation was used to generate the R2.

$$\text{R2} = 1 - \frac{TSSO}{SSEO}$$

$$\text{SSEO} = (1 - \text{R2}) \cdot TSSO$$

Where R2 is from any model and is calculated from the transformed TSSO and SSEO, TSSO is from the original data. The SSEO statistic shows what the sum of squared error would have been if the transformation had never taken place. The models with their equivalent SSEO values are listed in the table below.

<table>
<thead>
<tr>
<th>MODEL</th>
<th>SSEO</th>
<th>PARAMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIN</td>
<td>27676999</td>
<td>5</td>
</tr>
<tr>
<td>CD</td>
<td>17289542</td>
<td>5</td>
</tr>
<tr>
<td>NN1</td>
<td>32342910</td>
<td>6</td>
</tr>
<tr>
<td>BC0</td>
<td>25929477</td>
<td>5</td>
</tr>
<tr>
<td>BC1</td>
<td>16048453</td>
<td>6</td>
</tr>
<tr>
<td>BC2</td>
<td>16048404</td>
<td>7</td>
</tr>
<tr>
<td>CES</td>
<td>10013722</td>
<td>8</td>
</tr>
<tr>
<td>TRANS</td>
<td>15385819</td>
<td>9</td>
</tr>
<tr>
<td>NN2</td>
<td>26470145</td>
<td>12</td>
</tr>
<tr>
<td>QUAD</td>
<td>21336673</td>
<td>15</td>
</tr>
<tr>
<td>TLOG</td>
<td>15062963</td>
<td>15</td>
</tr>
<tr>
<td>NN3</td>
<td>16761550</td>
<td>18</td>
</tr>
<tr>
<td>NN4</td>
<td>13435483</td>
<td>24</td>
</tr>
</tbody>
</table>

The neural network models are labeled NN1, NN2, NN3, and NN4 with $k=1$, $k=2$, $k=3$, and $k=4$ in each respective model. With common units for the SSE of all models, likelihood ratio testing may proceed. The results of such testing follow.$^{4}$

$^{4}$ The likelihood ratio testing in this case takes place under the assumption that the likelihood functions are the same with the exception of the error values. In this way, not only are the transformed errors (SSE0) changed back to the original units (SSE0), the models which generated those errors are also transformed to the same footing. This is appropriate only if the error values are also transformed.
second is the unrestricted. The null hypothesis is that the SSE of the restricted model is lower than the SSE of the unrestricted model. This situation could not happen in traditional likelihood ratio testing with nested models. However, with non-nested models the possibility exists due to different functional specification. Additionally, REJECTED(DF=0) indicates the unrestricted model had a lower SSE than the restricted model, however, the number of parameters were the same in both cases. This is also impossible in traditional likelihood ratio testing, but in this context, it can happen.

In the listing the only model not rejected (or that no other model is accepted over) is NN4. This is an indication that the neural network model with k=4 degree of parameterization is preferable to all other models tested. The NN4 model exhibited a better fit than all other models tested. This result at conjunction with the result of White's test, indicating that the model does not over fit the sample, indicates the neural network model of the production function is preferable in this case.

**Concluding Remarks**

This paper has focused on the implementation of neural network techniques as nonparametric modeling techniques for production analysis. The argument began by showing neural networks to be universal approximators. Then a method for estimating neural network models using PROC NLIN was discussed. Finally, a test designed by White for optimal depth of parameterization was discussed. The test prevented over fitting of the sample due to excess parameters.

The results indicated the neural network model to be preferable to other parametric and flexible form models tested.

Areas for further research include other economic and noneconomic functional estimation. Virtually all mapping functions may be modeled by neural networks. If the phenomena is clearly understood, a nonparametric algorithm may be needlessly cumbersome. A parametric model that captures all aspects of an event cannot be improved upon by nonparametric methods. However, if there is doubt or incomplete information concerning the event in question, then nonparametric analysis techniques may be appropriate. Neural networks provide a workable modeling structure which can be fairly easily estimated. Additionally, and equally importantly, White's test provides
a solution to the bias/variance question inherent in nonparametric analysis.

SAS and SAS/IML are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. © indicates USA registration

Other brand and product names are registered trademarks or trademarks of their respective companies.

References


Hecht-Nielsen, R. (1990b), Neurocomputing. Addison-Wesley Publishing Company, Reading MA.


