AIChE

A New Weighted Optimal Combination of ANNs for Catalyst Design and Reactor Operation: Methane Steam Reforming Studies

Viswanathan Arcotumapathy, Arman Siahvashi, and Adesoji A. Adesina
Reactor Engineering and Technology Group, School of Chemical Engineering,
University of New South Wales, NSW 2052 Australia

DOI 10.1002/aic.12748
Published online September 20, 2011 in Wiley Online Library (wileyonlinelibrary.com).

Catalyst design and evaluation is a multifactorial multiobjective optimization problem and the absence of well-defined mechanistic relationships between wide ranging input-output variables has stimulated interest in the application of artificial neural network for the analysis of the large body of empirical data available. However, single ANN models generally have limited predictive capability and insufficient to capture the broad range of features inherent in the voluminous but dispersed data sources. In this study, we have employed a Fibonacci approach to select optimal number of neurons for the ANN architecture followed by a new weighted optimal combination of statistically-derived candidate ANN models in a multierror sense. Data from 200 cases for catalytic methane steam reforming have been used to demonstrate the veracity and robustness of the integrated ANN modeling technique. © 2011 American Institute of Chemical Engineers *AIChE J,* 58: 2412–2427, 2012

Keywords: steam reforming, weighted optimal combination, artificial neural network, catalyst design

Introduction

Artificial neural networks (ANNs) have been used in many chemical engineering systems to provide structure and clarity to large, multisource and multifactorial empirical data pools in order to secure usable quantitative models for descriptive, predictive and optimization purposes.[1] This is especially relevant for situations where detailed mechanistic knowledge sufficient to develop reliable mathematical models between input-output vectors is poor, ambiguous or even unavailable.[2] In particular, ANNs are better equipped to capture and manipulate nonlinear features in an input-output data set than empirical correlations.[3] As a result, ANN modeling and analysis has emerged as a versatile tool for data treatment and utilization in the nonlinear processes typically encountered in rational catalyst design and optimization as well as reactor engineering. For instance, Lemoine et al.[4] employed ANN analysis in the development of new, generalized and highly reliable correlations for predicting hydrodynamic and mass-transport parameters in multiphase reactors using data from widely dispersed sources. More recently, Chesterfield and Adesina[5] also obtained the functional relationship between titania photocatalyst preparation variables, its physiochemical properties and photoactivity using a 16-neuron single hidden layer feed forward ANN based on over 700 data cases from different laboratories over a 20-year period. Availability of the quantitative relationship between photocatalyst preparation variables such as calcination temperature, dopant ionic radius, oxidation number, etc., and the photoactivity for organic pollutant destruction has led to the design of new visible-light activated codoped titania for photocatalysis.[6]

In many ANN modeling studies, the optimal network architecture was often selected based on the candidate with the smallest value of one of a class of error indices, such as the residual sum of squares errors (SSE), the mean square error (MSE), the root-mean square error (RMSE), the mean absolute error (MAE), and the mean absolute percentage error (MAPE). However, since different error indices measure different extents and types of mismatch between the ANN model and the data, it is rare that a single ANN will have the smallest value for all five types of error indices from among a set of ANNs originally culled for consideration. In other situations, the selection of the optimal ANN, was based on a procedure in which the ANN with the best possible rank across all error indices is taken as the most superior network.[5] Further improvement in the prediction of the best estimate of the target (output) variable may be obtained by using a linear combination of ANNs satisfying an optimization criterion for any of the error indices. This method has been shown to give better estimates of the target value than of a single ANN. Arithmetic averaging, however, assumes that all error indices are equally important under all circumstances. Given that ANN is rooted in biomimicry (the science of replicating details which nature has perfected), it seems logical to optimize the number of neurons in the ANNs considered in this work using the Fibonacci approach—a number sequence often descriptive of many naturally-occurring optimal networks.[7–9]

Correspondence concerning this article should be addressed to A. A. Adesina at a.adesina@unsw.edu.au.

Table 1a. Eigenvalues of the Principal Components

Principal Component (PC)	Eigenvalue (EV)	% Variance	Cumulative % Variance
1	3.93049	28.07493	28.07493
2	3.040176	21.71554	49.790472
3	1.804639	12.89028	62.680752
4	1.397875	9.984821	72.665573
5	1.014833	7.24881	79.914383
6	0.862279	6.159139	86.073522
7	0.764299	5.459276	91.532798
8	0.534215	3.815819	95.348617
9	0.315637	2.254547	97.603164
10	0.204525	1.460892	99.064056
11	0.107554	0.768245	99.832301
12	0.016227	0.115904	99.948205
13	0.007251	0.051795	99.99999
14	1.53E-30	1.1E-29	100

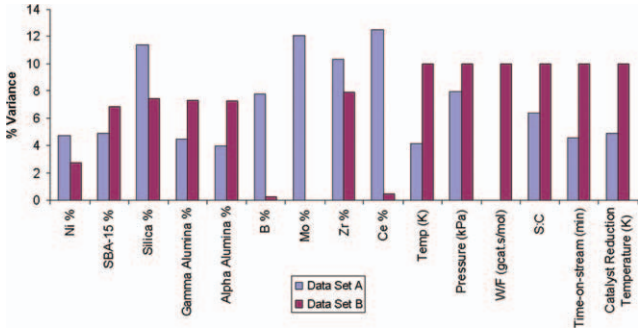

Figure 1. Percentage variance explained by principal components 1–8 for *Data Set A* and 1–10 for *Data Set B*.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

In this study, we introduce a weighted combination of ANNs with the smallest value for each of the various error indices in order to obtain the optimal estimate of the output vector. The weights assigned to contributing error indices are defined to take advantage of the information contained in ANNs with the upper and lower bounds of the particular error index in an orthogonal sense. In order to demonstrate the utility of the weighted optimal combination of ANNs, it has been applied to the evaluation of available data on the catalytic steam reforming of methane. The reaction is the most important route for the production of hydrogen or syngas[10] employed in the manufacturing of methanol, ammonia and the Fischer-Tropsch synthesis of hydrocarbons. Industrial reforming employs Ni-based catalyst. Exact formulation; however, vary due to the need for carbon deposition control and minimization. Catalyst design variables such as Ni loading, support type, for example SBA-15, SiO_2, α-Al_2O_3 and γ-Al_2O_3; promoters - B, Mo, Zr, Ce catalyst reduction temperature using H_2, as well as reactor operating factors such as steam-to-carbon (S:C) ratio W_{cat}/F_{CH_4}, reforming temperature and pressure to determine the overall product yield and conversion. Over 8,000 studies on various aspects of the catalytic steam reforming of methane within the past 10 years show that improvement in this key industrial processing technology is a multi-input-multi-output problem, which may be best handled using artificial neural network analysis of existing information library.[11–16] A multilayer feedforward network was used in this investigation in order to capture the diversity of experimental conditions and variability inherent in both data cases and the nature of the reaction itself. The ANN architecture is a layered structure; each layer has several neurons which receive inputs from neurons in the preceding layer, signal flows only in forward direction by the virtue of neurons connected between the layers. However, neurons are not connected within a layer. Hidden layers are present between input and output layer and the number can be increased to allow a network to extract higher-order statistics.[17]

Theoretical Basis and Numerical Procedure

Principal components and factor analysis

In multivariable systems data analysis, it is essential to investigate possible correlation between the explanatory (predictor) variables in the absence of response (output) variables as a first step in the modeling exercise. Indeed, pattern recognition under conditions with substantially high-dimensional input space may result in a model with suspect adequacy. Therefore, reduction of input dimensional space or examination of data for the presence of explanatory variable cluster(s) becomes important.[18,19] The $N \times pv$ matrix of predictor variables, where "N" represents the number of observations, and "pv" represents the number of predictor variables (sample observations representing the data cases cf. Table A1 of the Appendix) was standardized column wise, and the resulting matrix was used to find the covariance matrix, which was then used to derive the principal

Table 1b. Unrotated Factors

	Principal Component (PC)								
Variables	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	Communality
Ni %	0.292	−0.378	0.298	0.045	0.014	0.004	−0.013	0.240	0.377
SBA−15 %	0.173	0.507	0.132	0.068	0.015	0.028	0.235	−0.153	0.388
SiO_2 %	−0.049	−0.014	−0.066	0.391	0.819	−0.269	−0.097	−0.004	0.913
Gamma Alumina %	0.311	−0.307	−0.359	−0.042	−0.008	0.137	−0.024	0.139	0.360
Alpha Alumina %	−0.468	−0.001	0.008	−0.289	−0.003	0.026	−0.061	−0.098	0.316
B %	0.123	−0.055	−0.614	−0.077	0.003	0.266	0.016	−0.388	0.623
Mo %	0.206	−0.273	0.300	−0.027	−0.025	−0.208	−0.245	−0.808	0.965
Zr %	−0.062	0.029	−0.100	0.579	−0.571	−0.356	−0.158	0.027	0.828
Ce %	0.088	0.283	0.095	0.068	0.028	0.437	−0.834	0.098	0.999
Temperature (K)	0.368	0.193	−0.139	−0.345	−0.001	−0.134	0.001	0.020	0.329
Pressure (kPa)	−0.309	0.057	−0.067	−0.473	−0.004	−0.471	−0.292	0.073	0.639
S/C	−0.288	−0.169	0.377	−0.002	−0.009	0.460	0.188	−0.100	0.511
Time−on−stream (min)	0.142	0.525	0.130	0.014	0.013	−0.013	0.170	−0.142	0.363
Catalyst Reduction Temperature (K)	0.404	−0.049	0.302	−0.258	−0.011	−0.154	−0.025	0.202	0.389

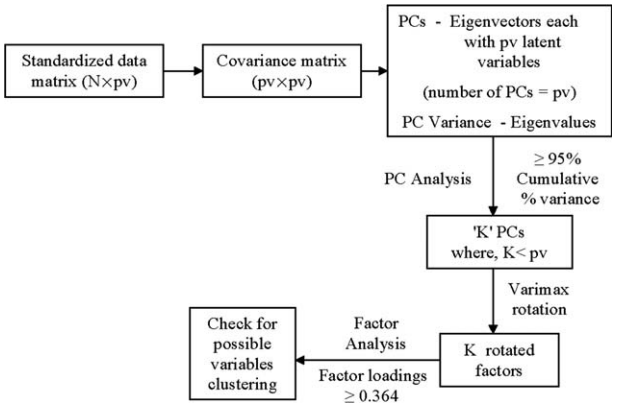

Figure 2. Principal components and factor analysis.

components (PCs). Each column of the eigenvector derived from the covariance matrix is known as a principal component, where the number of principal components is equal to the number of predictor variables which are uncorrelated, orthogonal and the coefficients (factor loadings) of a component are combined in a linear sense. The corresponding eigenvalues represent the variance associated with each of the components. This approach also ensures that errors within data sets used are critically screened and not carried over into the development and optimization of the ANNs.[18,20]

The minimum number of principal components required to summarize the variance contributed by each variable may be evaluated using one of three criteria, namely; Kaiser measure,[21] scree plot and cumulative percentage of variance. However, in this article, we have employed the cumulative percentage of variance criterion, because it is often more appropriate for engineering systems.[22] Additionally, the cumulative percentage variance of the selected components should be greater than or equal to 95%.[22] In view of the diversity in the range of factors used by different authors, the method proposed in this investigation has been applied to two major clusters of data containing varying mix of catalyst design and reactor engineering operating variables. Data Set A where relates to studies in which investigators explored mostly catalyst preparation and activation factors, while Data Set B, reflects the importance of reactor engineering variables such as space-time W_{cat}/F_{CH_4}, steam:carbon ratio, pressure and temperature among others.

Data Set A

It is evident that the principal components 1–8, in Table 1a are significant since the cumulative percentage variance is 95.35%. Thus, Ni loading, SBA-15, SiO_2, α-Al_2O_3, γ-Al_2O_3, B, Mo, Zr, Ce, reforming temperature, pressure, S:C ratio, time-on-stream and catalyst reduction temperature are the predictors considered. Therefore, the first eight principal components (cf. in Table 1b) were used to derive the percentage variance contributed by each variable to the explanatory data matrix. The common variance contributed by a variable is given by its communality which is the sum of squares of factor loadings for that variable across components. The variance contributed by each of the variables is shown in Figure 1. It is evident from this plot that, α-alumina at 4%, and cerium at 12.5% contributed the minimum and maximum variance, respectively. To identify the presence of significant possible correlations among variables the principal components were rotated. The flow chart in Figure 2 shows the procedure involved in deriving the principal components and factor analysis through components rotation. To enhance interpretation, the selected principal components (i.e., PCs 1–8) were subjected to orthogonal rotation using the Varimax technique which retains the uncorrelated principal components. The critical absolute value of correlation coefficient between the factor loadings and the rotated factor (i.e., rotated principal component) is 0.364 at 99% confidence level. Therefore, to retain a variable for further investigation, the corresponding factor loading should be $\geq$ 0.364 on one factor only. The highlighted factor loadings for the rotated factors shown in Table 1c indicate that all the variables are important in explaining the total variance in the explanatory data set. In particular, there is no evidence of variable clusters because a factor can only be considered for variable cluster analysis either if, there are three variables with factor loadings $\geq$ 0.8 or four factor loadings $\geq$ 0.6.[20]

Data Set B

The explanatory variables from these data set consisted of reforming temperature, pressure W_{cat}/F_{CH_4}, S:C ratio, time-on-stream Ni loading, SBA-15, SiO_2, α-Al_2O_3, γ-Al_2O_3, B, Zr, Ce and catalyst reduction temperature. It required PCs 1 to 10 with a cumulative variance of 99.8% to derive the percentage variance contributed by each variable to the explanatory data matrix (cf. Figure 1). A minimum variance of 0.2% was contributed by boron to the data matrix, while a

Table 1c. Rotated Factors

	Principal Component (PC)							
Variables	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Ni %	0.274	−0.261	**0.474**	−0.057	−0.014	0.029	0.015	−0.067
SBA−15 %	0.120	**0.608**	−0.026	−0.027	−0.016	−0.049	0.021	0.007
SiO_2 %	0.048	−0.013	−0.017	−0.065	**0.952**	−0.007	0.002	−0.015
Gamma Alumina %	0.326	**−0.386**	−0.112	−0.106	−0.079	−0.263	0.006	0.077
Alpha Alumina %	**−0.478**	−0.067	−0.147	−0.127	−0.085	0.195	0.007	0.004
B %	0.198	−0.130	**−0.684**	−0.165	−0.106	−0.205	0.005	−0.132
Mo %	−0.019	0.010	0.006	0.021	0.014	0.016	−0.001	**−0.982**
Zr %	0.072	−0.025	−0.025	**0.903**	−0.071	−0.007	0.003	−0.023
Ce %	0.005	−0.003	−0.001	−0.002	−0.002	0.001	**−0.999**	−0.001
Temperature (K)	−0.003	0.136	0.041	−0.180	−0.130	**−0.509**	0.006	−0.014
Pressure (kPa)	**−0.722**	−0.137	0.080	0.024	0.020	−0.302	0.001	−0.015
S/C	0.033	0.009	0.048	−0.252	−0.180	**0.641**	0.005	−0.037
Time−on−stream (min)	0.035	**0.594**	−0.014	−0.024	−0.016	−0.085	−0.016	0.002
Catalyst Reduction Temperature (K)	0.068	0.020	**0.511**	−0.154	−0.119	−0.280	−0.006	−0.079

Table 2. Comparing Models by Stepwise Backward Variable Selection for Data Set A and B

Model	Variables included in the model	Mallow's C_p	
		Data Set A	Data Set B
M1	v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14		
M2	v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14	13.0	10.1
M3	v1, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14	13.0	10.1
M4	v1, v2, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14	13.0	10.1
M5	v1, v2, v3, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14	13.0	10.1
M6	v1, v2, v3, v4, v6, v7, v8, v9, v10, v11, v12, v13, v14	13.0	10.1
M7	v1, v2, v3, v4, v5, v7, v8, v9, v10, v11, v12, v13, v14	13.0	10.1
M8	v1, v2, v3, v4, v5, v6, v8, v9, v10, v11, v12, v13, v14	13.0	10.1
M9	v1, v2, v3, v4, v5, v6, v7, v9, v10, v11, v12, v13, v14	13.0	10.1
M10	v1, v2, v3, v4, v5, v6, v7, v8, v10, v11, v12, v13, v14	13.0	41.5
M11	v1, v2, v3, v4, v5, v6, v7, v8, v9, v11, v12, v13, v14	131.8	10.1
M12	v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v12, v13, v14	61.5	9.4
M13	v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v13, v14	91.6	82.0
M14	v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v14	12.3	12.4
M15	v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13	32.9	14.7

maximum contribution was observed from all the reactor operation variables and catalyst reduction temperature at 10% each. The subsequent rotation of 10 PCs revealed that all the variables can be retained for further analysis and there is no evidence of variable clustering. As a prelude to ANN modeling, multilinear regression analysis is necessary in order to determine possible multicolinearity in the regressor variables during output prediction in order to have a more optimal set of input variables for the ANN model.

Multiple linear regression approach

The stepwise backward variable selection method involves application of multilinear model using all 14 regressor variables with the elimination of one variable at a time, followed by the reduced model evaluation on the basis of various discriminating criteria. The Mallows criterion (C_{pr})[23] was employed since it is generally considered more efficient than others such as F-test.[20] For a reduced model C_{pr} is calculated using

$$C_{pr} = \frac{SSE_{pr}}{\hat{\sigma}^2} - N + 2pr \tag{1}$$

where, SSE_{pr} is the sum-of-squared errors of reduced model with p variables, $\hat{\sigma}^2$ is the variance present in the full model, N is the number of observations, pr is the number of variables in the reduced model, and C_{pr} is the Mallows criterion.

A reduced model is considered to be a valid model if the associated $C_{pr} \approx pr$.

Data Set A. The multilinear regression analysis was carried out with the regressor variables, Ni loading (v1), SBA-15 (v2), SiO_2 (v3), α-Al_2O_3 (v4), γ-Al_2O_3 (v5), B (v6), Mo (v7), Zr (v8), Ce (v9), reforming temperature (v10), pressure (v11), S:C ratio (v12), time-on-stream (v13), and catalyst reduction temperature (v14) with methane conversion as the response variable. $C_{pr} \approx 13$ for the reduced models M2 to M10 (cf. in Table 2) suggests that variables v1 to v9 are equally important, and there is also evidence of multicollinearity among some of these variables,[22] in particular between the variables v1 to v9 and v10 to v14.

Data Set B. The regressor variables from these data set; Ni loading (v1), SBA-15 (v2), SiO_2 (v3), α-Al_2O_3 (v4), γ-Al_2O_3 (v5), B (v6), Zr (v7), Ce (v8), reforming temperature (v9), pressure (v10), (v11), S:C ratio (v12), time-on-stream (v13), and catalyst reduction temperature (v14) were regressed against methane conversion. From Table 2, it is evident that the exclusion of variables v1 to v8 and v10 to build reduced models M2 to M9 and M11, respectively, resulted in same value of $C_{pr} \approx 10$. Consequently, there is proof of multicollinearity among variables, namely variables v1 to v8 and v10 with variables v11 to v14 and v9. In the presence of high degree of multicollinearity in the data, multilinear regression becomes redundant for building parsimonious models[24] that can be used for predictive and descriptive purposes and, hence, the adoption of artificial neural network analysis for a more rigorous model development.

Artificial neural network approach

The input variables comprised of Ni, support types- SBA-15, SiO_2, α-Al_2O_3, γ-Al_2O_3, promoters- B, Mo, ZrO_2, CeO_2, catalyst reduction temperature, reforming temperature, pressure, steam:carbon (S:C) ratio, and time-on-stream. All these variables are in different units and thus, vary by orders of magnitude. The entire input data was therefore recalibrated so that all the values fall into the interval [0, 1]. The data scaling can shorten the learning time of the network.[25] If x represents original value for a variable then

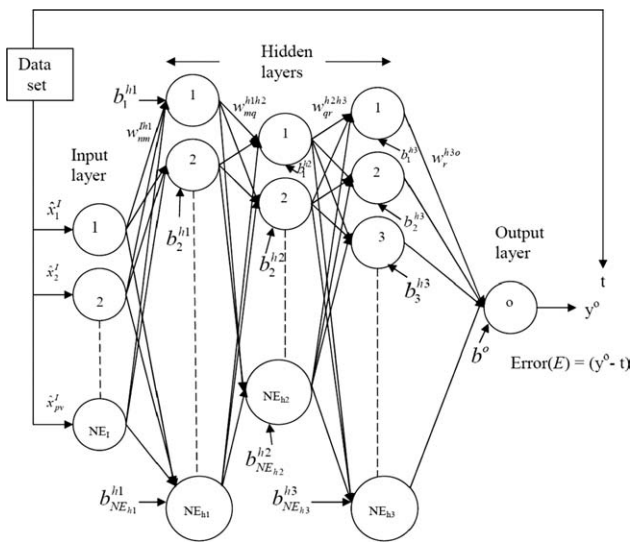

Figure 3. A multilayer feedforward neural network with three hidden layer.

$$\hat{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{2}$$

where $\hat{x}$ is the rescaled variable, $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the original independent variable, respectively. The response variable was similarly rescaled over the interval [0, 1].

Selection of the optimum number of neurons and network architecture

The method employed to select the optimum number of neurons for building the multilayer feedforward neural network is shown in Figure 3. The feedforward network with supervised learning algorithms like backpropagation algorithms is competent in learning the relationship between the given inputs and targets for a nonlinear process data set.[5] The numerical procedure was carried out in MATLAB neural network toolbox™ version 7.8.0.347 (R2009a). The stepwise procedure involved in selecting optimum number of neurons and implementation of the same to obtain the optimum network topology is detailed in our previous work.[26] This investigation is based on two data sets—A and B—as earlier indicated. For each set, the data was divided into 75%, 12.5%, and 12.5% for training, validation and testing of the networks, respectively.

In order to accommodate criteria proposed by Haykin[17] and Barron,[27] we adopted the rule of thumb, $1 \leq N_{hl}/N_{tr} \leq 10$.[5] Since 150 and 70 data cases were chosen for training the networks from *Data Sets A* and *B*, respectively, the final term of the Fibonacci series employed should not exceed this value, in these cases $N_{hl} \leq 1{,}500$ (i.e., 15 ANNs), and 700 (i.e., 14 ANNs), respectively. Levenberg-Marquardt algorithm (cf. Appendix) was used to train each network based on the randomly selected data cases. The training includes the ''early stopping'' procedure which avoids overfitting of the data to the model.[28] Quite frequently, no single ANN will have the smallest value for all error indices across the board, thus further refinement was sought through a combination of ANNs with varying types of minimal/maximal error indices. This has led to the development of (1) mean-squared error optimal linear combination (MSE-OLC) method.[29] This approach has produced significant improvement in neural network analysis as demonstrated for catalyst design and thermal machine tooling.[30–32] However, the method presumes that MSE alone (among all other error indices) is sufficient to accommodate all the variability in the data. In this study, we relax this assumption in two ways. First, we posit that superior improvement may be achieved by use of a multiple-error-based optimal linear combination (ME-OLC) in which only the neural networks with the minimum error indices (including *MSE*) across the set are combined in the same manner as the MSE-OLC. In the second approach, we introduce the additional requirement that each of the ANNs with the minimal error indices do not necessarily contribute equally to final choice, but do so in a manner proportional to the fraction of the error index with respect to the sum of all error indices at its best and worst values. Consequently, a new multiple-error-based weighted optimal combination (ME-WOC) is introduced. The basis for the optimal ANN combination in each case is recapitulated to aid understanding of the computational procedure.

Combining Trained Neural Networks

MSE-OLC approach

In neural network modeling, trained networks are evaluated on the basis of the error indices (viz; *SSE, MSE, RMSE, MAE, MAPE*), however, only ANNs with the relatively low MSEs are considered in output prediction. Consequently, the optimum output vector

$$\underline{y}^{opt}_{MSE-OLC} = \sum_{j=0}^{cn_{MSE-OLC}} \alpha_j \underline{y}_j \tag{3}$$

where *cn* is the number of ANNs whose *MSE* satisfy in the following condition given

$$\frac{MSE - MSE_{\min}}{MSE_{\max} - MSE_{\min}} \leq 0.1 \tag{4}$$

within the set of ANNs, $cn_{MSE\text{-}OLC} < S_{\text{ANN}}$. Additionally, the coefficient vector $= [\alpha_j]^T$ is obtained from the relation $\alpha = \psi^{-1}$, where $\underline{\psi}$ is a $(cn_{MSE\text{-}OLC} + 1) \times (cn_{MSE\text{-}OLC} + 1)$ matrix whose elements are given by

$$\underline{\psi}_{(i+1),(j+1)} = \underline{y}_i \underline{y}_j \tag{5}$$

where, $i, j = 0, 1, 2, \ldots\ldots, cn_{MSE\text{-}OLC}$ with $\underline{\psi}_{1,1} = 1$ (i.e., $<\underline{y}_0$ is a $(1 \times N)$ unit vector, and $\underline{y}_i$ or $\underline{y}_j$ are the output of component networks, thus, $\underline{\psi}$, is a symmetric matrix given as

$$\underline{\psi}_{(cn_{MSE-OLC}+1)\times(cn_{MSE-OLC}+1)} = \begin{bmatrix} \underline{y}_0\underline{y}_0 & \underline{y}_0\underline{y}_1 & \underline{y}_0\underline{y}_2 \cdots\cdots\cdots & \underline{y}_0\underline{y}_{cn_{MSE-OLC}} \\ \underline{y}_1\underline{y}_0 & \underline{y}_1\underline{y}_1 & \underline{y}_1\underline{y}_2 \cdots\cdots\cdots & \underline{y}_1\underline{y}_{cn_{MSE-OLC}} \\ \vdots & \vdots & \vdots \ddots & \vdots \\ \underline{y}_m\underline{y}_0 & \underline{y}_m\underline{y}_1 & \underline{y}_m\underline{y}_m \cdots\cdots\cdots & \underline{y}_m\underline{y}_{cn_{MSE-OLC}} \\ \vdots & \vdots & \vdots \ddots & \vdots \\ \underline{y}_{cn_{MSE-OLC}}\underline{y}_0 & \underline{y}_{cn_{MSE-OLC}}\underline{y}_1 & \underline{y}_{cn_{MSE-OLC}}\underline{y}_2 \cdots\cdots & \underline{y}_{cn_{MSE-OLC}}\underline{y}_{cn_{MSE-OLC}} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \underline{y}_0\underline{y}_1 & \underline{y}_0\underline{y}_2 \cdots\cdots\cdots\cdots & \underline{y}_0\underline{y}_{cn_{MSE-OLC}} \\ \underline{y}_1\underline{y}_0 & \underline{y}_1^2 & \underline{y}_1\underline{y}_2 \cdots\cdots\cdots\cdots & \underline{y}_1\underline{y}_{cn_{MSE-OLC}} \\ \vdots & \vdots & \vdots \ddots & \vdots \\ \underline{y}_m\underline{y}_0 & \underline{y}_m\underline{y}_1 & \underline{y}_m^2 \cdots\cdots\cdots & \underline{y}_m\underline{y}_{cn_{MSE-OLC}} \\ \vdots & \vdots & \vdots \ddots & \vdots \\ \underline{y}_{cn_{MSE-OLC}}\underline{y}_0 & \underline{y}_{cn_{MSE-OLC}}\underline{y}_1 & \underline{y}_{cn_{MSE-OLC}}\underline{y}_2 \cdots\cdots & \underline{y}_{cn_{MSE-OLC}}^2 \end{bmatrix} \tag{6}$$

and U, is a $(cn_{MSE\text{-}OLC} + 1) \times 1$ vector defined

$$\underline{U} = \underline{t}\,\underline{y}_i \tag{7}$$

Table 3. Ranking of Networks Based on Two-Way ANOVA

ANN Number of neurons in hidden layer	Error Indices			Error Index-based ranking			Gross rank	Position of the gross rank	MAE
		SSE	RMSE	MAE	SSE	RMSE			
1	2.6876	0.0985	0.0738	13	14	1	28	10	
2	1.8558	0.0802	0.0578	12	13	14	39	14	
3	1.1848	0.0672	0.0488	10	10	13	33	12	
5	0.8544	0.0581	0.0401	7	8	12	27	9	
8	0.429	0.0453	0.03	1	3	9	13	4	
13	0.4565	0.0456	0.0289	3	4	7	14	5	
21	0.5949	0.0465	0.029	6	6	8	20	6	
34	0.4619	0.0435	0.0264	4	2	6	12	2	
55	**0.4509**	**0.0432**	**0.0251**	**2**	**1**	**4**	**7**	**1**	
89	5.5746	0.0658	0.0392	14	9	11	34	13	
144	0.9491	0.0523	0.0257	9	7	5	21	7	
233	0.9057	0.0673	0.038	8	11	10	29	11	
377	0.4762	0.0462	0.0189	5	5	2	12	3	
610	8.3323	0.1584	0.092	15	15	15	45	15	
987	1.511	0.0749	0.0207	11	12	3	26	8	

where $i = 0, 1, 2, 3,\ldots\ldots..$ $cn_{MSE\text{-}OLC}$, and t is the target vector for the performance variable.

ME-OLC approach

The ME-OLC introduced in this study uses the same optimal output as in the *MSE-OLC*, but differs from the latter in the choice of the selected list of ANNs and, therefore, defined as

$$\underline{y}^{opt}_{ME\text{-}OLC} = \sum_{j=0}^{cn_{ME\text{-}OLC}} \lambda_j \underline{y}_j \tag{8}$$

where $cn_{ME\text{-}OLC}$ is the number of ANNs with the smallest value for each error index as distinct from *MSE-OLC* where the selected networks are those with relatively low *MSE* (cf. Eq. 4). Moreover, since *RMSE* and *SSE* are related to the MSE, the three error indices considered are, *MSE, MAE* and *MAPE* (i.e., $cn_{ME\text{-}OLC} = 3$). However, λ_j are chosen exactly as α_j (i.e., $\underline{\lambda} = \underline{\psi}^{-1}\underline{U}$). Clearly both ψ and U are (4×4) matrix, and (4×1) vector, respectively with elements defined as described in Eqs. 5 and 7, where $i, j = 0, 1, 2, 3$.

ME-WOC approach

In both the *MSE-OLC* and *ME-OLC*, the optimal output $\underline{y}^{opt}$, was obtained by considering members of the ANN set characterized by some of the smallest estimates of the error indices. However, since the set of ANNs was initially chosen such that the total number of neurons N^{opt} in all hidden layers is the same for each ANN, then even ANN members, with relatively poor estimates of the error indices contain information about the data set that can be used to fine-tune the optimal output to be as close as possible to the target value. This is akin to the ecological notion that even the weakest member of an animal group contributes in some sense to the overall optimal functionality of the commune. Consequently, the optimal output vector $\underline{y}^{opt}_{WOC}$ in this approach, was chosen to reflect the influence of ANN members with the smallest and worst error estimates for each index via appropriate weighting factors as provided later.

ME-WOC algorithm

1 The minimum $(w_{EI\text{-}min})^p$ and maximum $(w_{EI\text{-}max})^p$ local weights for error indices MSE, MAE and MAPE were calculated for the pth iteration using Eqs. 9 and 10, respectively

$$w^p_{EI-\min} = \left[\frac{EI - \min}{\sum_{j=1}^{cn_{woc}} EI_j}\right] \tag{9}$$

$$w^p_{EI-\max} = \left[\frac{EI - \max}{\sum_{j=1}^{cn_{woc}} EI_j}\right] \tag{10}$$

where

EI: error index (i.e., MSE or MAE or MAPE)

$EI - \min$: smallest value for any EI

$EI - \max$: highest value for any EI

cn_{woc}: number of component ANNs of WOC

p: iteration counter (p = 1, 2, 3,……..)

2. The minimum $(\beta_{EI\text{-}min})^p$ and maximum $(\beta_{EI\text{-}max})^p$ global weights for error indices MSE, MAE and MAPE were calculated for the pth iteration as given in Eqs. 11 and 12, respectively

$$\beta^p_{EI-\min} = \frac{w^p_{EI-\min}}{\sum_{i=1}^{3} w^p_{EI-\min_i}} \tag{11}$$

$$\beta^p_{EI-\max} = \frac{w^p_{EI-\max}}{\sum_{i=1}^{3} w^p_{EI-\max_i}} \tag{12}$$

3. Calculate the weighted optimal $(\underline{y}^{opt}_{woc})^p$ and worst $(\underline{y}^{worst}_{woc})^p$ combination of ANNs for the pth iteration as per Eqs. 13 and 14, respectively.

$$\left(\underline{y}^{opt}_{woc}\right)^p = \sum_{i=1}^{3} \beta^p_{EI-\min_i} \underline{y}_{EI-\min_i} \tag{13}$$

$$\left(\underline{y}^{worst}_{woc}\right)^p = \sum_{i=1}^{3} \beta^p_{EI-\max_i} \underline{y}_{EI-\max_i} \tag{14}$$

where

$\underline{y}_{EI-\min}$: output of ANN with minimum error index

$\underline{y}_{EI-\max}$: output of ANN with maximum error index

4. Compute the ratio (ϕ^p_{EI}) of error indices MSE, MAE and MAPE for the pth iteration

$$\varphi^p_{EI} = \frac{\sum \left| \{ |\left(\underline{y}^{opt}_{woc}\right)^p| - |\left(\underline{y}^{worst}_{woc}\right)^p| \} \right|}{\sum \left| \{ |\underline{y}_{EI-\min}| - |\underline{y}_{EI-\max}| \} \right|} \tag{15}$$

Table 4. Ranking of Networks Based on Two-Way ANOVA

ANN Number of neurons in hidden layer	Error Indices		Error Index-based ranking		Gross rank	Position of the gross rank
	MAE	MAPE	MAE	MAPE		
34	0.0306	0.0204	51	53	104	51
35	0.0288	0.0191	41	41	82	42
36	0.0288	0.0191	39	42	81	41
37	0.0268	0.0180	18	26	44	23
38	0.0267	0.0179	16	23	39	18
39	0.0301	0.0193	50	47	97	49
40	0.0260	0.0172	10	11	21	10
41	0.0298	0.0195	48	49	97	50
42	0.0279	0.0186	33	34	67	33
43	0.0329	0.0212	53	54	107	54
44	0.0278	0.0185	32	33	65	32
45	0.0278	0.0184	31	31	62	31
46	0.0337	0.0217	55	55	110	55
47	0.0260	0.0176	9	16	25	11
48	0.0286	0.0190	36	40	76	37
49	0.0266	0.0177	15	18	33	16
50	0.0294	0.0194	45	48	93	45
51	0.0269	0.0178	20	20	40	19
52	0.0313	0.0203	52	52	104	52
53	0.0287	0.0187	38	36	74	36
54	0.0273	0.0181	24	29	53	27
55	0.0298	0.0193	47	46	93	46
56	0.0291	0.0189	44	39	83	43
57	0.0295	0.0196	46	50	96	48
58	0.0288	0.0192	40	44	84	44
59	0.0276	0.0183	28	30	58	30
60	0.0271	0.0180	22	27	49	25
61	0.0251	0.0166	5	5	10	5
62	0.0275	0.0181	27	28	55	29
63	0.0264	0.0173	13	13	26	12
64	0.0329	0.0203	54	51	105	53
65	0.0289	0.0187	43	35	78	38
66	0.0269	0.0179	19	24	43	22
67	0.0285	0.0192	35	43	78	39
68	0.0270	0.0178	21	19	40	20
69	0.0274	0.0177	25	17	42	21
70	0.0262	0.0171	12	8	20	9
71	0.0354	0.0228	56	56	112	56
72	0.0281	0.0188	34	37	71	35
73	0.0267	0.0171	17	9	26	13
74	0.0266	0.0172	14	12	26	14
75	0.0250	0.0164	4	3	7	3
76	0.0259	0.0171	8	7	15	7
77	0.0257	0.0171	6	10	16	8
78	0.0275	0.0178	26	22	48	24
79	0.0277	0.0179	29	25	54	28
80	0.0261	0.0173	11	15	26	15
81	0.0300	0.0192	49	45	94	47
82	0.0286	0.0185	37	32	69	34
83	**0.0241**	**0.0161**	**1**	**1**	**2**	**1**
84	0.0258	0.0169	7	6	13	6
85	0.0245	0.0164	3	4	7	4
86	0.0271	0.0173	23	14	37	17
87	0.0288	0.0188	42	38	80	40
88	0.0277	0.0178	30	21	51	26
89	0.0244	0.0164	2	2	4	2

5. If $\boldsymbol{\phi}^p_{MSE}$, ϕ^p_{MAE}, and $\phi^p_{MAPE}) \leq 1$ then stop, go to step 9.
6. Update minimum and maximum local weights for error index (EI) MSE, MAE and MAPE

$$w^{p+1}_{EI-\min} = w^p_{EI-\min}[1 + \varphi^p_{EI}] \tag{16}$$

$$w^{p+1}_{EI-\max} = w^p_{EI-\max}[1 + \varphi^p_{EI}] \tag{17}$$

7. Increment the iteration counter, p = p + 1.
8. Go to step 2.
9. Terminate the procedure.

Relative importance of steam methane reforming (input) variables

Although all input variables, identified via PCA are significant determinants in the estimation of the output values for the optimal ANN, the relative importance of each input variables is required in order to understand the role of the individual inputs in the analysis of the optimal ANN. Garson[33] proposed a method which has been widely used[34–36] for evaluating the relative importance of each input variable.

However, in order to extract additional insights from the neural network learning process, alternatives to the evaluation of the importance of input variables has been proposed.[37–39] Olden et al.[38] proposed a connection weight method (CWM), and compared it against other evaluative methods such as the Garson, partial derivatives, input perturbation, sensitivity analysis, forward stepwise addition, backward stepwise elimination and improved stepwise selection methods and concluded that the CWM is significantly more accurate in procuring the relative importance between input-to-output than all others. A significant virtue of the CWM is that the results indicate whether an input variable has a detrimental or favorable influence with respect to the output. The connection weights (CW) for a typical three-hidden layer model (as shown in Figure 3) are given by Eqs. 18–20. However, the connection weights from inputs to 1st hidden layer (w^{Ih1}_{nm}), 1st hidden to 2nd hidden layer (w^{h1h2}_{mq}), 2nd hidden to 3rd hidden layer (w^{h2h3}_{qr}) and 3rd hidden to output layer (w^{h3o}_{r}) are raw weights obtained from trained, validated, tested and simulated best ANN.

For inputs and 2nd hidden layer

$$cw^{Ih2}_{nq} = \sum_{m=1}^{NE_{h1}} w^{Ih1}_{nm} w^{h1h2}_{mq} \tag{18}$$

For inputs and 3rd hidden layer

$$cw^{Ih3}_{nr} = \sum_{q=1}^{NE_{h2}} cw^{Ih2}_{nq} w^{h2h3}_{qr} \tag{19}$$

For inputs and output layer

$$cw^{Io}_{n} = \sum_{r=1}^{NE_{h3}} cw^{Ih3}_{nr} w^{h3o}_{r} \tag{20}$$

where I, $h1$, $h2$, $h3$ and o represents input, 1st hidden, 2nd hidden, 3rd hidden and output layers, respectively, and $n = 1, 2,\ldots\ldots\ldots\ldots.., NE_I$ (number of neurons in the input layer which is same as the number of variables), $m = 1, 2,\ldots\ldots\ldots., NE_{h1}$ (number of neurons in 1st hidden layer), $q = 1, 2,\ldots\ldots\ldots., NE_{h2}$ (number of neurons in 2nd hidden layer), $r = 1, 2,\ldots\ldots\ldots., NE_{h3}$ (number of neurons in 3rd hidden layer), $o = 1$ (number of neurons in output layer).

Results and Discussion

Data Set A

Selection of Optimum Number of Neurons. A total of 200 data cases from various studies[11–16] dealing with catalyst design and evaluation for methane steam reforming is analyzed using the procedure outlined. The optimum number of neurons required in the hidden layer(s) of the ANN model to achieve significant model accuracy is carried out by training 15 single hidden layer feed-forward ANN's. These networks were distinguishable by the number of neurons in the

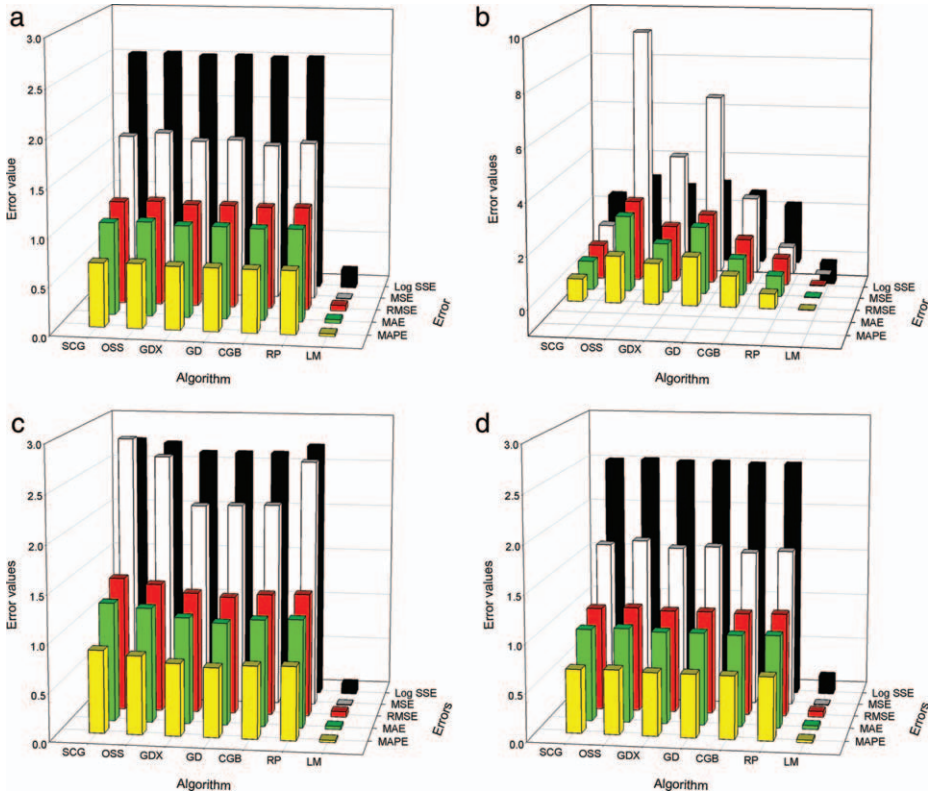

Figure 4. a) Average log SSE, MSE, RMSE, MAE and MAPE of networks combined. b) Average log SSE, MSE, RMSE, MAE and MAPE of single-hidden layer networks combined. c) Average log SSE, MSE, RMSE, MAE and MAPE of two-hidden layer networks combined. Average log SSE, MSE, RMSE, MAE and MAPE of three-hidden layer networks combined.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

hidden layer, namely; 1, 2, 3, 5, 8, 13, 21, 34, 55, 89 144, 233, 377, 610 and 987 (members of the Fibonacci series). The average *SSE, MSE, RMSE, MAE* and *MAPE* for each network with a varying number of neurons was calculated following the algorithm prescribed in the earlier section on "Methodology to select the optimum number of neurons and network architecture".

The analysis of variance of *SSE, MSE, RMSE, MAE* and *MAPE* against different networks (i.e., against total number of neurons in the hidden layer) revealed that only *SSE, RMSE* and *MAE* are significant. The network model with 55 neurons in the hidden-layer ranked 1st based on the significant errors as shown in Table 3.

Consequently, the implementation of step 2, requires that $x = 34$ and $z = 89$, resulting in 56 new ANNs with different number of neurons, in the single hidden layer, viz, 34, 35, 36,……., 89. A two-way *ANOVA* of the ANNs (56) vs. error index (5) revealed that *MAE* and *MAPE* were the only statistically significant error indices. Further treatment confirmed that the ANN with 83 neurons would be the "best" choice (cf. Table 4).

Optimization of ANN architecture

Given that $NE^{opt} = 83$ neurons a new set of ANNs with one-, two- and three-hidden layers were generated as outlined in step 3 resulting in 3404 different ANNs distributed as one single-hidden layer ANN, 82 two-hidden layer ANNs, and 3321 three-hidden layer ANNs. The optimization algorithm for the ANN may be carried out using a variety of techniques including Resilient backpropagation (RP), Conjugate gradient with Powell/Beale restarts (CGB), gradient descent (GD), gradient descent with momentum and adaptive learning rate (GDX), one-step secant (OSS), scaled conjugative gradient (SCG) and Levenberg-Marquardt (LM). As may be seen in Figure 4a–d the LM technique gave the most efficient function approximation for each of the error indices and networks considered. This finding is consistent with the study of Hagan and Menhaj[40] and has been adopted as the optimization algorithm in subsequent data treatment in this work. The 3404 ANNs were subsequently evaluated based on the *MSE-OLC, ME-OLC* and *ME-WOC* approaches.

Modes of optimal ANN combination

The results of the simulation for the 3404 networks showed that three-hidden layer ANNs generally exhibited superior performance to the single- and two-hidden layer networks. The apparent smaller average error indices for the single- and two-hidden layer architecture displayed in Figure 4a–e is due to the relatively low ratio of the single:three-hidden layer networks that is, 1:3321, and the two:three-hidden layer networks, i.e., 82:3321.

The nearest three-hidden layer ANNs are those with 14-23-46; 30-28-25 and 21-5-57 in the 1st-, 2nd-, 3rd-hidden layers, respectively. The associated MSEs were 0.00020, 0.00023 and 0.00024, respectively. Figure 5b shows the parity between the predicted output (methane conversion) and the companion target. The results for the ME-OLC approach are also displayed in Figure 5c, while 5d confirms the suitability of the ME-WOC approach for the accurate prediction

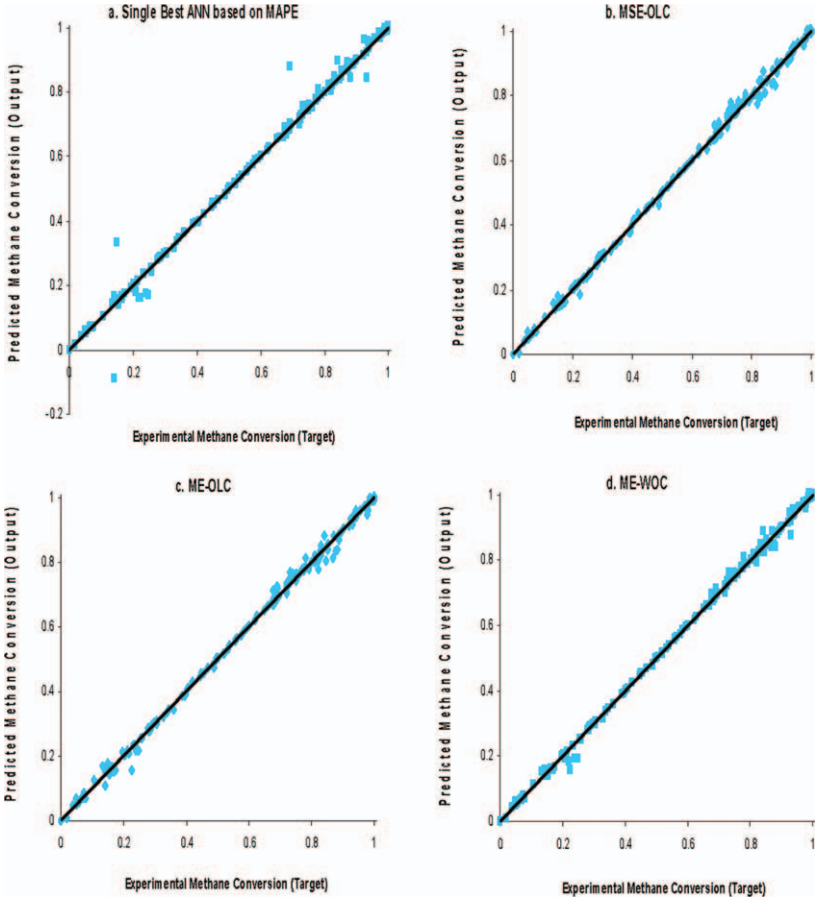

Figure 5a–d. Parity plots for single best and combinations of ANN models.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

of target values and the respective ANN models used for these combinations are listed in Table 5.

Although visual comparison of Figure 5a–d show that the optimal combination of ANNs gave better estimates of the output, it is difficult to discriminate between the three different modes of the ANN combination (Figure 5b–d) on the basis of R^2 statistics ($R^2 > 0.99$ in each case) only. However, normal probability plot may be used to effect a more refined selection since it can reveal whether or not a model sufficiently explains the variability present in the data with respect to the response.[41] Figure 6a–6d display the normal probability plots for the optimal ANN, *MSE-OLC, ME-OLC* and *ME-WOC* models. All four plots clearly indicate that the nonlinear data had undergone transformation to a linear standard data set via ANN treatment, suggesting that the optimal ANN model (in each case) can be used for descriptive, predictive and control purposes. Even so, it is evident that the standardized errors for the majority of the data points lie closer to the normalized line for *ME-WOC* than those for either the *MSE-OLC* or *ME-OLC* or optimal ANN models. It may be noted that the width of *AB* for *ME-WOC* is larger with the pivotal point "0" on the *x*-axis.

Even though *MSE-OLC* is pivoted on the "0" point, the width AB is smaller compared to *ME-WOC*. While for *ME-OLC* and optimal ANN the width of AB is smaller in contrast to *ME-WOC* and not pivoted on the "0" point of *x*-axis.

The behavior of the error indices (especially *MSE, MAE* and *MAPE*) for the *ME-WOC* method as a function of the iteration is portrayed in Figure 7a–c. *MAE* experienced a characteristic hyperbolic drop followed by a rise to an iteration-invariant value. While *MAPE* showed a gentle decline and *MSE* increased linearly with iteration. On the basis of

Table 5. ANN Models Employed for ME-OLC and ME-WOC Combination

	ANN					
Model characteristics	1st Hidden Layer neurons	2nd Hidden Layer neurons	3rd Hidden Layer neurons	MSE	MAE	MAPE
*#Least MAPE	17	44	22	0.00083	0.00978	0.00596
*#Least MSE	14	23	46	0.00020	0.00881	0.00685
*#Least MAE	30	28	25	0.00023	0.00880	0.00673
#Highest MSE, MAE & MAPE	45	1	37	3.68203	1.89951	1.25209

* Denotes component ANN model considered for ME-OLC combination.
# Denotes component ANN model considered for ME-WOC combination.

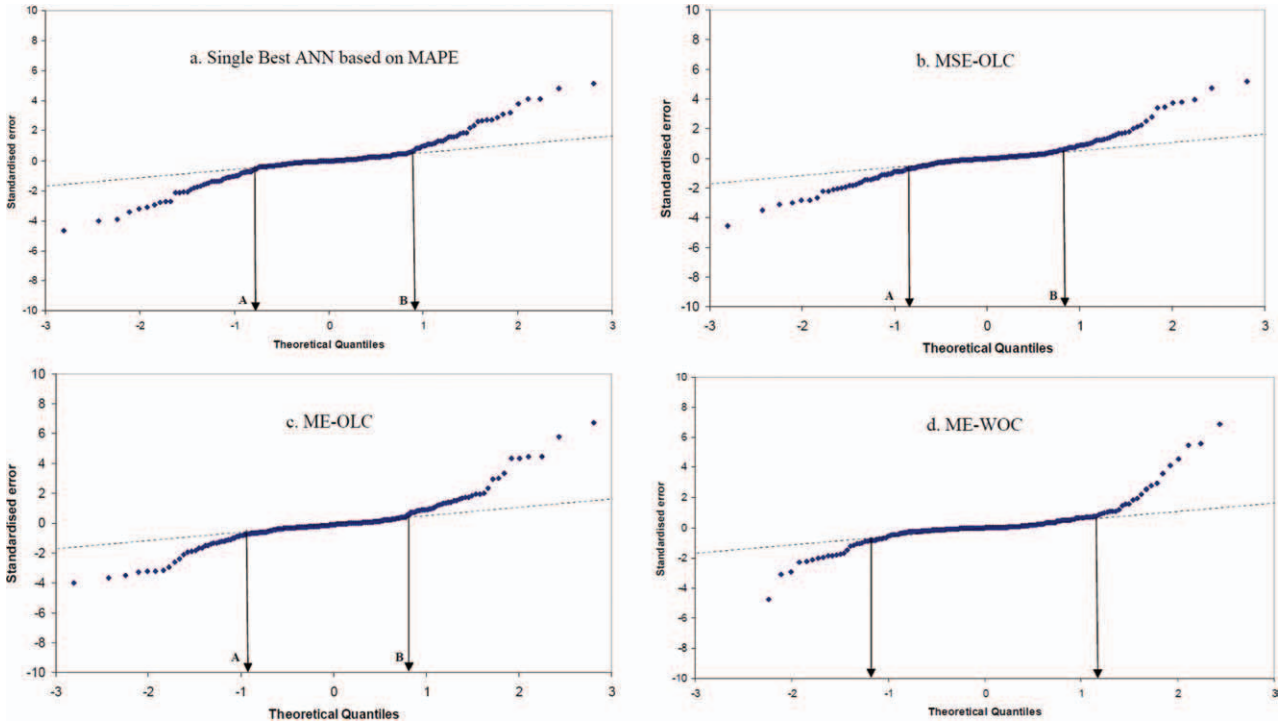

Figure 6a–d. Normal Probability plots for single best and combinations of ANN models.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

these features, *MAPE* offers the best capture of variation present in the data pool than the other error indices. Consequently, *MAPE* was used as the basis error index to compare optimal single and combination of ANN models as illustrated in Figure 8.

Data Set B

Selection of the optimum number of neurons and ANN architecture

A total of 94 data cases from methane steam reforming studies[11–13,16] was analyzed using the procedure outlined in Arcotumapathy et al.[26] Step 1 of the modeling involved training 14 single-hidden layer feedforward ANN's, distinguishable by the number of neurons in the hidden layer, namely; 1, 2, 3, 5, 8, 13, 21, 34, 55, 89 144, 233, 377 and 610 (members of the Fibonacci series). The analysis of variance of SSE, MSE, RMSE, *MAE* and *MAPE* against different networks (i.e., against total number of neurons in the hidden layer) revealed that only *SSE, RMSE, MAE* and MAPE are significant. The network model with 13 neurons in the hidden-layer ranked 1st on the basis of significant errors as shown in Table 6. Consequently, the implementation of Step 2, requires that $x = 8$ and $z = 21$, resulting in 14 new ANNs with different number of neurons, in the single hidden layer, viz, 8, 9, 10,……..19, 20, 21. A two-way *ANOVA* of the ANNs (14) vs. error index (5) revealed that *SSE, RMSE* and *MAE* were the only statistically significant error indices. Further treatment confirmed that the ANN with 9 neurons would be the "best" choice (cf. Table 7). With $NE^{opt} = 9$ neurons a new set of 37 ANNs of one-, two- and three-hidden layers were generated. The two-hidden layer ANN with 7 and 2 neurons in the first and second layer, respectively, was found to be superior with minimum value across all error indices namely; *SSE, MAE* and *MAPE* are 0.0174, 0.0085, 0.0087, respectively.

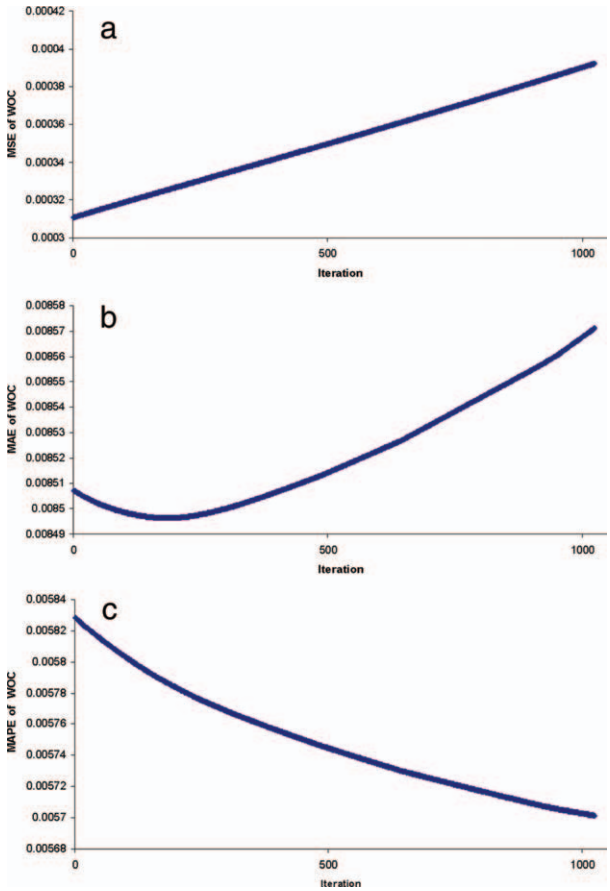

Figure 7. a) MSE values against iterations. b) MAE values against iterations. c) MAPE values against iterations.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

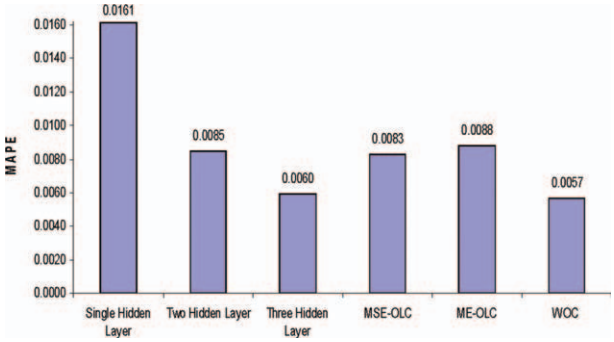

Figure 8. Comparing MAPE of different individual and combined ANN models.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Interpretation of neural networks

Data Set A. The weights from best model were used to obtain a relationship between the input variables to output by employing Olden's connection weight method and the results are schematically represented in Figure 9. The most influential factor with respect to methane conversion was found to be time-on-stream followed by pressure, temperature, mesoporous support SBA-15, catalyst reduction temperature, SiO_2, Ni, boron, cerium, α-alumina, zirconium, γ-alumina, steam-to-methane ratio and Mo. It is also evident from the neural network learning that the reactor operating variables, viz; reforming temperature (773–1023 K), pressure (0.1–1 MPa) and time-on-stream (> 40 h) have relatively high importance compared to the other variables for catalyst design.

Time-on-stream had the largest connection weight and is detrimental to methane conversion. This ANN analysis points to decreased catalyst activity with reforming time in agreement with earlier studies.[11,13–16] An increase in operating variable, S:C ratio (i.e., steam:methane ratio between 2–5.5), with the smallest connection weight increases methane conversion.[11,14,15]

Catalyst evaluation via ANN modeling suggests that when Ni loading (in the range 5–60.9%) in a catalyst system is increased, there is a decrease in methane conversion which suggests that for a given catalyst support and promoter combination, the optimum *Ni* loading is most likely at, or below, 5 wt %.[11–16] Among the supports compared in this study SBA-15 appears to be the best which is evident from Wan et al.[12] study where, methane conversion of 99.8% was observed using *Ni* (5% loading) supported on SBA-15. Boron and cerium were found to be equally effective as promoters in the range 0.5–1 wt % and 5–10 wt %, respectively. Catalyst reduction temperature appears to have a positive influence on methane conversion and this phenomenon was observed in a study carried out by Matsumura and Nakamori[16] where the catalyst reduced at 973 K was substantially more active than that treated at 773 K.

Data Set B. Parity between the optimum ANN output and experimental methane conversion is shown in Figure 10 and the connections weights between the input variables and the methane conversion can be found in Figure 11. Similar to the findings in Data Set A, time-on-stream was found to be most influential factor followed by the mesoporous support SBA-15, reaction temperature, catalyst reduction temperature, pressure, cerium, steam-to-methane ratio, zirconium, SiO_2, γ-alumina, α-alumina, space-time (W_{cat}/F_{CH_4}), boron and Ni. It seemed that the inclusion of space-time, improved the relative importance of reaction temperature and catalyst reduction temperature over reactor operating pressure. The analysis revealed that a modest (7%) increase in reaction temperature resulted in a four-fold increase in methane conversion, while the data showed only a linear relationship between methane conversion and W_{cat}/F_{CH_4}. The latter finding is consistent with the kinetics in a reactor system with minimal external transport effects. This suggests that for practical purposes, reaction temperature control is more critical than feed flow rate regulation. Nonetheless, on the basis of catalyst composition, higher Ni % percentage has a detrimental effect on methane conversion, while, SBA-15 and Ce appears to be the most important support and promoter, respectively.

We further remark that, while ANN modeling is a quantitative approach to the treatment of a multifactorial, large data pool, it does not provide explicit functional relations that can be used for calculus-based optimization. However, as a numerical optimization strategy for large data systems, it provides an estimation of the magnitude and direction of the optimum values for discrete and/or qualitative variables that play key roles in determining the system performance. For example, while there is no known mathematical relation between methane conversion and support type (a clear weakness if calculus-based optimization method was to be used),

Table 6. Ranking Based on Two-Way ANOVA

ANN Number of neurons in hidden layer	Error Indices				Error Index-based ranking				Gross rank	Position of the gross rank
	SSE	RMSE	MAE	MAPE	SSE	RMSE	MAE	MAPE		
1	1.101168	0.086196	0.059553	0.067396	10	12	11	10	43	10
2	0.438353	0.061844	0.03962	0.039545	4	6	9	9	28	7
3	0.774876	0.063212	0.036308	0.037604	8	8	8	8	32	8
5	0.506764	0.055484	0.029005	0.026901	5	4	5	4	18	5
8	0.321501	0.048553	0.028683	0.028511	1	2	4	5	12	3
13	**0.333663**	**0.048311**	**0.026867**	**0.026115**	**2**	**1**	**3**	**3**	**9**	**1**
21	0.350167	0.050429	0.0248	0.022924	3	3	2	2	10	2
34	0.691916	0.062115	0.03132	0.029679	7	7	6	6	26	6
55	1.105947	0.069164	0.035235	0.035105	11	9	7	7	34	9
89	0.54319	0.057842	0.024682	0.022128	6	5	1	1	13	4
144	1.518369	0.085217	0.059946	0.079231	13	11	12	12	48	12
233	1.491678	0.086401	0.063298	0.087041	12	13	14	13	52	14
377	1.99882	0.098001	0.058805	0.078457	14	14	10	11	49	13
610	1.033905	0.069164	0.061077	0.094188	9	10	13	14	46	11

Table 7. Ranking Based on Two-Way ANOVA

ANN Number of neurons in hidden layer	Error Indices			Error Index-based ranking			Gross rank	Position of the gross rank
	SSE	RMSE	MAE	SSE	RMSE	MAE		
8	0.379771	0.049955	0.027682	2	2	6	10	3
9	**0.372436**	**0.049146**	**0.026854**	**1**	**1**	**3**	**5**	**1**
10	0.536083	0.05254	0.028742	11	5	9	25	8
11	0.42231	0.0505	0.026564	3	3	1	7	2
12	0.430176	0.050559	0.027145	4	4	4	12	4
13	0.466859	0.053065	0.028537	8	7	8	23	7
14	0.447254	0.05337	0.028806	6	9	10	25	8
15	0.611285	0.055793	0.029042	13	12	11	36	12
16	0.434314	0.052578	0.026773	5	6	2	13	5
17	0.564299	0.056801	0.029691	12	13	14	39	13
18	0.51426	0.054647	0.029224	9	10	12	31	11
19	0.458056	0.053186	0.027169	7	8	5	20	6
20	0.533886	0.055274	0.028393	10	11	7	28	10
21	0.620161	0.058131	0.029318	14	14	13	41	14

ANN clearly points to the superiority of SBA-15 (among all the supports examined in the studies represented by either of the two data sets employed in this work) and provides quantitative information on the size and direction of its influence. Additionally, even for quantitative factors, the predicted optimum magnitude and direction (decreasing or increasing) for each of the input variables is sufficient for design purposes. Thus, in this case, the optimum Ni and promoter content were found to be 5 wt % and 0.5–1 wt % (depending on promoter type), respectively. On the other hand, increasing the reaction temperature within the range, 773–1023 K, is desirable but moderated by catalyst reduction temperature. While this is consistent with the thermodynamics of an endothermic reaction (e.g., steam reforming) and the optimum temperature progression path (based on the rate-temperature-conversion behavior) for a reversible reaction, the findings from ANN analysis revealed that thermodynamics and kinetic information alone is not sufficient for comprehensive process reactor operation since catalyst reduction temperature, reactor space-time and time-on-stream have quantified significant roles as indicated earlier.

Conclusions

In this study, a systematic approach to the selection of optimum number of neurons and number of hidden layers in ANN model building and optimization routed in the Fibonacci technique has been espoused. Intrusive noise in the primary data used for the training, validation and testing of the ANN models was minimized using PCA to screen out factors which may lead to redundancy. Moreover, an early stopping procedure was implemented during ANN training to further eliminate overfitting and vouchsafe the reliability of the inferences from model prediction and optimization. Thus, despite the diversity in data source and time over which they were generated, ANN models with excellent learning rate and performance were derived. Specifically, the optimum number of neurons required to capture the input-output behavior was first obtained following a Fibonacci algorithm. The ANN architecture was subsequently optimized resulting in the generation of "best" ANN candidates based on common error indices. The prediction of target output values was then secured through a multierror based weighted combination of the emergent "best" ANNs. This technique was demonstrably superior to the predictive performance of optimal ANN rooted in a single-error index using the large data pool for methane reforming. The findings were not only consistent with qualitative evidence, but also surprisingly offered insights into the relative importance of some hitherto unacknowledged factors which could now serve as stimulus for improvement in the methane (or hydrocarbon) reforming reactor technology and catalyst design. In particular, the addition of boron, cerium or related group metals (at less than 1 wt %) to Ni-based

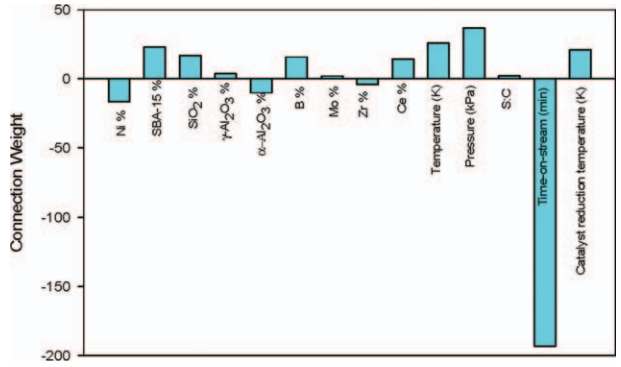

Figure 9. Connection weights of variables against methane conversion (*Data Set A*).

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

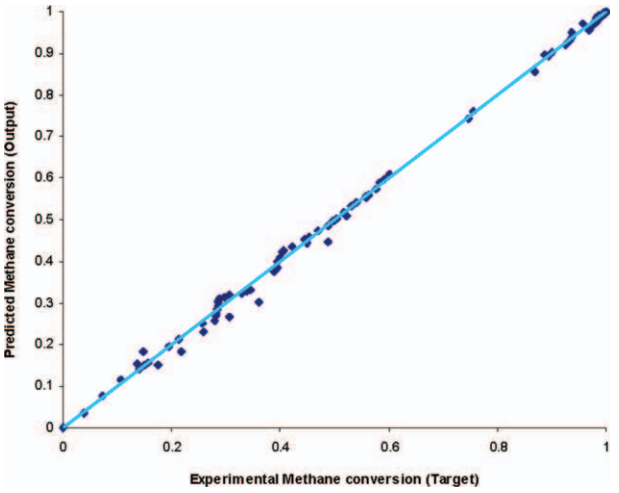

Figure 10. Parity plot for single best ANN model (*Data Set B*).

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

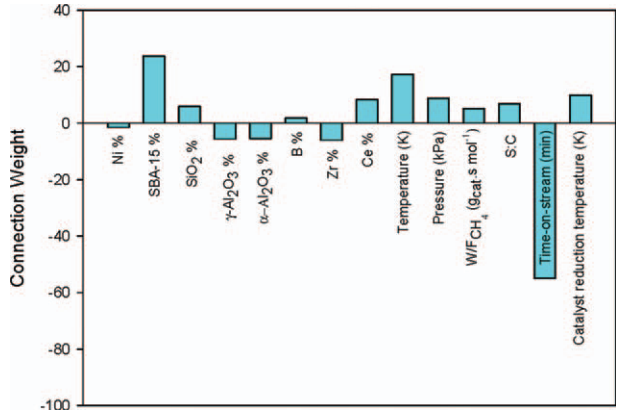

Figure 11. Connection weights of variables against methane conversion (*Data Set B*).

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

catalysts, and the appearance of SBA-15 as a superior support to γ-alumina was identified.

Indeed, reactor operating variables were found to offer better prospects than the manipulation of catalyst design factors for improving reforming technology.[26] The application of the optimal linear or weighted combination of various error indices (*ME-OLC* and *ME-WOC*) to other multi-input-multi-output data systems in chemical engineering are self-evident particularly because of the underlying advantage of the Fibonacci method in the determination of the most efficient number of neurons for the ANN architecture.

Notation

α = combination weights of component ANN models for MSE-OLC
β = global weights for component ANN models for ME-WOC
δ = Marquardt sensitivity
λ = combination weights of component ANN models for ME-OLC
μ = adjustable parameter
$\hat{\sigma}^2$ = variance of full model
ϕ = ratio of the sum of absolute difference in weighted optimum and weighted worst outputs to the sum of absolute difference in best and worst outputs of ANN models based on given error index, EI
ψ = correlation matrix of component ANN outputs with constant vector for MSE-OLC and ME-OLC
ANN = artificial neural network
b = bias
C = Mallows criterion
cn = component networks of combination model
CW = connection weights
CWM = connection weights method
E = error
EI = error index
F = molar flow rate, moles.s^{-1}
exp = exponential
h = hidden layer
I = identity matrix
J = Jacobian
M = model
MAE = mean absolute error
MAPE = mean absolute percentage error
ME = multierror
MSE = mean squared error
N = total number of observations or data cases
NE = neurons
OLC = optimum linear combination
PC = principal component
Pr = number of variables in reduced model
pv = total number of variables
RMSE = root mean squared error
S = possible combination of ANN architectures for given number of neurons and hidden layers
SSE = sum-of-squared errors
st = number of data cases in training data set
stt = number of data cases in testing data set
sv = number of data cases in validation data set
t = targets
U = product of target vector and output vectors including constant vector for MSE-OLC and ME-OLC
W = weight in grams (or) artificial neural network weights
WOC = weighted optimal combination
x = original variable (or) inputs to neuron
$\hat{x}$ = scaled variable
y = output

Subscripts

cat = catalyst
EI = error index
EI-max = maximum error index
EI-min = minimum error index
*h*1 = first hidden layer (or) hidden layer
*h*2 = second hidden layer
*h*3 = third hidden layer
i = summation or multiplication counter
I = input layer
j = summation or multiplication counter
m = first hidden layer neuron (or) hidden layer neuron
max = maximum
min = minimum
n = input layer neuron
NE = number of neurons
pr = reduced model
q = second hidden layer neuron
r = third hidden layer neuron
s = data case
test = testing
tr = training
valid = Validation

Superscripts

O = output layer
opt = optimum
p = iteration number
T = transpose of a vector or matrix

Literature Cited

1. Pham DT. *An introduction to artificial neural networks*. In: Bulsari AB. *Neural Networks for Chemical Engineers*. Amsterdam: Elsevier; 1995:1–19.
2. Bishop C. *Neural Network for Pattern Recognition*. Oxford: University Press; 1995:1–28.
3. Behkish A, Lemoine R, Oukaci R, Morsi BI. Novel correlations for gas holdup in large-scale slurry bubble column reactors operating under elevated pressures and temperatures. *Chem Eng J*. 2006;115:157–171.
4. Lemoine R, Behkish A, Sehabiague L, Heintz YJ, Oukaci R, Morsi BI. An algorithm for predicting the hydrodynamic and mass transfer parameters in bubble column and slurry bubble column reactors. *Fuel Process Tech*. 2008;89:322–343.
5. Chesterfield D, Adesina AA. Evidence-based design and optimization of titania photocatalyst via artificial neural network analysis. *J Chem Eng Jpn*. 2009;42:s185–s198.
6. Hudaya T, Qazaq AS, Lucien FP, Adesina AA. Design of $Ce_{y}Co_{x}Ti_{(1-x)}O_{3+\delta}$ Perovskite for Photocatalysis: A Statistical Study. *Adv Oxidat Tech*. 2009;12:16–28.
7. Bejan A, Lorenete S, Lee J. Unifying constructal theory of tree roots, canopies and forests. *J Theor Biol*., 2008;254:529–540.
8. Arneodo A, Argoul F, Muzy JF, Tabard M. Uncovering Fibonacci sequences in the fractal morphology of diffusion-limited aggregates. *Phys Lett*. 1992;171:31–36.

9. Stakhov AP. The generalized principle of Golden Section and its applications in mathematics, science and engineering. *Chaos Solitons Fractals*. 2005;26:263–269.
10. Simpson AP, Lutz AE. Exergy analysis of hydrogen production via steam methane reforming. *Int J Hydrogen Energy*. 2007;3:4811–4820.
11. Hou K, Hughes R. The kinetics of methane steam reforming over a Ni/α-Al_2O catalyst. *Chem Eng J*. 2001;82:311–328.
12. Wan H, Li X, Ji S, Huang B, Wang K, Li C. Effect of Ni Loading and $Ce_xZr_{1-x}O_2$ Promoter on Ni-Based SBA-15 Catalysts for Steam Reforming of Methane. *J Nat Gas Chem*. 2007;16:139–147.
13. Xu J, Chen L, Tan KF, Borgna A, Saeys M. Effect of boron on the stability of Ni catalysts during steam methane reforming. *J Catal*. 2009;261:158–165.
14. Numaguchi T, Shoji K, Yoshida S. Hydrogen effect on α-$A1_20_3$ supported Ni catalyst for steam methane reforming reaction. *Appl Catal A*. 1995;133:241–262.
15. Maluf SS, Assaf EM. Ni catalysts with Mo promoter for methane steam reforming. *Fuel*. 2009;88:1547–1553.
16. Matsumura Y, Nakamori T. Steam reforming of methane over nickel catalysts at low reaction temperature. *Appl Catal A*. 2004;258:107–114.
17. Haykin S. *Neural Networks: A Comprehensive Foundation*. 2nd ed. New Jersey: Prentice Hall; 1999.
18. Wang F. *Factor analysis and principal-components analysis*. In: Kitchin R, Thrift N. *International Encyclopedia of Human Geography*. Oxford: Elsevier; 2009:1–7.
19. Jobson JD. *Applied Multivariate Data Analysis: Volume II: Categorical and Multivariate Methods*. New York: Springer-Verlag; 1992.
20. Stevens J. *Applied Multivariate Statistics for the Social Sciences*. 3rd ed. New Jersey: Lawrence Erlbaum Associates; 1996.
21. Yidana SM, Ophori D, Banoeng-Yakubo B. A multivariate statistical analysis of surface water chemistry data-The Ankobra Basin, Ghana. *J Environ Manage*. 2008;86:80–87.
22. Hair JF, Black WC, Babin BJ, Anderson RE. *Multivariate Data Analysis: A Global Perspective*. 7th ed. Upper Saddle River, NJ: Prentice Hall; 2010.
23. Mallows CL. Some comments on C_p. *Technometrics*. 1973;15: 661–675.
24. Martin EB, Morris AJ, Zhang J. Artificial neural networks and multivariate statistics. In: Bulsari AB. *Neural networks for chemical engineers*. Amsterdam: Elsevier; 1995:627–658.
25. Tsaptsinos D. *Back-propagation and its variations*. In: Bulsari AB. *Neural Networks for Chemical Engineers*. Amsterdam: Elsevier, 1995:33–75.
26. Arcotumapathy V, Alenazey F, Adesina AA. Artificial neural network modeling of forced cycling operation between propane steam reforming and CO_2 carbon gasifier. *Catal Today*. 2011;164:275–281.
27. Barron AR. Universal approximation bounds for superpositions of a sigmodal function. *IEEE Trans Info Theory*. 1993;39:930–945.
28. Hagan MT, Demuth HB, Beale MH. *Neural Network Design*. Boston: PWS; 1995.
29. Hashem S. Optimal linear combinations of neural networks. *Neural Networks*. 1997;10:599–614.
30. Tomposa A, Margitfalvi JL, Tfirst E, Végvári L. Evaluation of catalyst library optimization algorithms: Comparison of the holographic research strategy and the genetic algorithm in virtual catalytic experiments. *Appl Catal A*. 2006;303:72–80.
31. Cundari TR, Deng J, Zhao Y. Design of a propane ammoxidation catalyst using artificial neural networks and genetic algorithms. *Ind Eng Chem Res*. 2001;40:5475–5480.
32. Yan J, Yang J. Optimal linear combination of neural networks to model thermally induced error of machine tools. *Int J Model Ident Contr*. 2009;8:139–147.
33. Garson GD. Interpreting neural-network connection weights. *AI Expert*. 1991;6:47–51.
34. Pareek VK, Brungs MP, Adesina AA, Sharma R. Artificial neural network modeling of a multiphase photodegradation system. *J Photochem Photobiol A Chem*. 2002;149:139–146.
35. Chen J, Chen W. Diagnostic analysis of a small-scale incinerator by the Garson index. *Infor Sci*. 2008;178:4560–4570.
36. Aleboyeh A, Kasiri MB, Olya ME, Aleboyeh H. Prediction of azo dye decolorization by UV/H_2O_2 using artificial neural networks. *Dyes Pigments*. 2008;77:288–294.
37. Yoon Y, Guimaraes T, Swales G. Integrating artificial neural networks with rule-based expert systems. *Decision Support Sys*. 1994;11:497–507.
38. Olden JD, Joy MK, Death RG. An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. *Ecol Model*. 2004;178:389–397.
39. Hattori T, Kito S. Analysis of factors controlling catalytic activity by neural network. *Catal Today*. 2006;111:328–332.
40. Hagan MT, Menhaj MB. Training Feedforward Networks with the Marquardt Algorithm. *IEEE Trans Neural Networks*. 1994;5:989–993.
41. Montgomery DC, Runger GC, Hubele NF. *Engineering statistics*. 3rd Ed. New York: Wiley; 2003.

Appendix

Artificial neural network training based on Levenberg-Marquardt algorithm

This training procedure is based on an illustrative ANN model schematically represented in Figure 3.

1. All weights and biases for a network are initialized by random number generation between [−1, 1].

2. The inth input neuron with input $\hat{x}_{in}$ from the training data set has an output

$$y^I_{n_s} = \hat{x}^I_{in_s} \tag{A1}$$

where

in = 1, 2, …, pv and pv is the number variables

n = 1, 2,……, NE_I and NE_I is the number of input neurons (which is equal to number of variables, i.e., NE_I = pv), and s represents the data case number.

3. The input to m^{th} neuron in the first hidden layer is given by the sum of weighted summation of all outputs from input layer and bias related to that neuron

$$x^{h1}_{m_s} = \left(\sum_{i=1}^{NE_I} w^{Ih1}_{nm} y^I_{n_s}\right) + b^{h1}_m \tag{A2}$$

where, m = 1, 2, …, NE_{h1} represents the number of neurons in the first hidden layer (h1) and b^{h1}_m denotes the bias applied to the first hidden layer neuron m.

4. A tan sigmoid function is applied to all the hidden layer neurons. Therefore, the output from the m^{th} neuron in the first hidden layer is given by

$$y^{h1}_{m_s} = \frac{2}{1+\exp(-2x^{h1}_{m_s})} - 1 \tag{A3}$$

5. Similarly, for the second hidden layer the input and output for q^{th} neuron is given by

$$x^{h2}_{q_s} = \left(\sum_{m=1}^{NE_{h1}} w^{h1h2}_{mq} y^{h1}_{m_s}\right) + b^{h2}_q \tag{A4}$$

$$y^{h2}_{q_s} = \frac{2}{1+\exp(-2x^{h2}_{q_s})} - 1 \tag{A5}$$

here, q = 1, 2, …, NE_{h2} denotes the number of neurons in the second hidden layer (h2) and b^{h3}_r represents the bias applied to the hidden neuron q.

6. Similarly, for the third hidden layer the input and output for the r^{th} neuron is given by

$$x^{h3}_{r_s} = \left(\sum_{q=1}^{NE_{h2}} w^{h2h3}_{qr} y^{h2}_{q_s}\right) + b^{h3}_r \tag{A6}$$

$$y^{h3}_{r_s} = \frac{2}{1+\exp(-2x^{h3}_{r_s})} - 1 \tag{A7}$$

where, r = 1, 2, …, NE_{h3} represents the number of neurons in the third hidden layer (h3) and denotes the bias applied to the hidden neuron r.

7. The input to a output layer neuron is given as

$$x_s^o = \left(\sum_{r=1}^{NE_{h3}} w_r^{h3o} y_{r_s}^{h3} \right) + b^o \tag{A8}$$

where 'o' denotes the single output neuron present in the output layer and represents the bias applied to the output neuron.

8. The overall output of the network is given by

$$y_s^o = x_s^o \tag{A9}$$

A linear transfer function is used in the output neuron therefore the input and output are the same.

9. Every neuron in each layer excluding the input layer has a sensitivity based on the network error (i.e., $(y_s^\circ - t_s)$). The Marquardt sensitivity terms for the neurons are calculated starting from neurons in the output layer to first hidden layer, since LM is a backpropagation algorithm. Therefore, the Marquardt sensitivity (δ_s°) for the lone neuron in the output layer is calculated

$$\delta_s^o = -2(y_s^o - t_s) \tag{A10}$$

10. Similarly the sensitivity for a neuron "r" in the third hidden layer (h3) is given by

$$\delta_{r_s}^{h3} = [1 - (y_{r_s}^{h3})^2] w_r^{h3o} \delta_s^o \tag{A11}$$

11. Similarly, for a neuron "q" in the second hidden layer (h2) is given by

$$\delta_{q_s}^{h2} = [1 - (y_{q_s}^{h2})^2] [\underline{w}_q^{h2h3}]^T \underline{\delta}_s^{h3} \tag{A12}$$

underlined notations represent a matrix or a vector. Where, $\underline{w}_q^{h2h3}$ is a vector of weights from qth neuron in the second hidden layer to all the neurons in the third hidden layer (i.e., it is a vector of ($1 \times NE_{h3}$) elements) and $\underline{\delta}_s^{h3}$ is a vector of all the sensitivity terms relative to the neurons in the third hidden layer (i.e., it is a vector of ($NE_{h3} \times 1$) elements).

12. Sensitivity for a neuron "m" in the first hidden layer (h1) is given by

$$\delta_{m_s}^{h1} = [1 - (y_{m_s}^{h1})^2] [\underline{w}_m^{h1h2}]^T \underline{\delta}_{q_s}^{h2} \tag{A13}$$

where, $\underline{w}_m^{h1h2}$ is a vector of weights from mth neuron in the first hidden layer to all the neurons in the second hidden layer (i.e., it is a vector of ($1 \times NE_{h2}$) elements) and $\underline{\delta}_s^{h2}$ is a vector of all the sensitivity terms relative to the neurons in the second hidden layer (i.e., it is a vector of ($NE_{h2} \times 1$) elements).

13. Steps 2 to 12 are repeated for all the data cases in the entire training data set.

14. The sum-of-squared error (SSE_{tr}) of the training set, for the p[th] iteration is calculated and stored in the memory which is given by

$$SSE_{tr}^p = \sum_{s=1}^{st} \left(E_{tr_s}^p \right)^2 \tag{A14}$$

where, $E_{tr_s}^p = \left(y_s^o - t_s \right)^p$ is the error for s[th] data case in the training data set for the p[th] iteration and s = 1, 2, …, st is the number of data cases in the training set and «t» is the scaled targets.

15. Similarly, steps 2 to 8 are repeated for all the data cases in validation set. The p[th] iteration mean squared error (MSE_{valid}) for validation set is calculated and stored in the memory which is given by:

$$MSE_{valid}^p = \frac{1}{sv} \sum_{s=1}^{sv} \left(E_{valid_s}^p \right)^2 \tag{A15}$$

where, $E_{valid_s}^p = \left(y_s^o - t_s \right)^p$ and s = 1, 2, …, sv is the number of data cases in the validation set.

16. The MSE_{valid}^p value and the corresponding weights and biases are stored in the memory as MSE_{valid} (minimum), weights (minimum) and biases (minimum), respectively in the memory.

17. Similarly, steps 2 to 8 are repeated for all the data cases in testing data set. The p[th] iteration mean squared error (MSE_{test}) for testing set is calculated and stored in the memory which is given by

$$MSE_{test}^p = \frac{1}{stt} \sum_{s=1}^{stt} \left(E_{test_s}^p \right)^2 \tag{A16}$$

where, $E_{test_s}^p = \left(y_s^o - t_s \right)^p$ and s = 1, 2, …, stt is the number of data cases in the testing set.

18. The weights and biases are adjusted using a numerical optimization technique based on Levenberg-Marquardt algorithm, for instance if $\underline{x}^p$ is a vector of weights and biases of the neural network for the p[th] iteration, then the weights and biases are updated for next iteration as given below

$$\underline{x}^{p+1} = \underline{x}^p - \left[J^T(\underline{x}^p) J(\underline{x}^p) + \mu I \right]^{-1} J^T(\underline{x}^p) \underline{E}_{tr}^p \tag{A17}$$

μ: An adjustable parameter used to achieve the required approximation (initial value, $\mu = 0.01$).

I: Identity matrix

$J(\underline{x}^p)$: Jacobian matrix (i.e., partial derivatives of network errors for all the data cases $\underline{E}_{tr}^p$ in the training data set for p[th] iteration with respect to weights and biases of the network). Therefore, each row of the Jacobian corresponds to network error obtained for that particular input/output data case, for example the row of Jacobian related to error $E_{tr_s}^p$ of s[th] data case is given as

$$\left[\begin{array}{l}
\left(\frac{\partial E_{tr_s}^p}{\partial w_{1,1}^{h1}} \; \frac{\partial E_{tr_s}^p}{\partial w_{2,1}^{h1}} \; \cdots \; \frac{\partial E_{tr_s}^p}{\partial w_{NE_I,1}^{h1}} \right) \cdots \left(\frac{\partial E_{tr_s}^p}{\partial w_{1,NE_{h1}}^{h1}} \; \cdots \; \frac{\partial E_{tr_s}^p}{\partial w_{NE_I,NE_{h1}}^{h1}} \right) \left(\frac{\partial E_{tr_s}^p}{\partial b_1^{h1}} \; \frac{\partial E_{tr_s}^p}{\partial b_2^{h1}} \; \cdots \frac{\partial E_{tr_s}^p}{\partial b_{NE_{h1}}^{h1}} \right) \\
\left(\frac{\partial E_{tr_s}^p}{\partial w_{1,1}^{h1h2}} \; \frac{\partial E_{tr_s}^p}{\partial w_{2,1}^{h1h2}} \; \cdots \; \frac{\partial E_{tr_s}^p}{\partial w_{NE_{h1},1}^{h1h2}} \right) \cdots \left(\frac{\partial E_{tr_s}^p}{\partial w_{1,NE_{h2}}^{h1h2}} \; \cdots \; \frac{\partial E_{tr_s}^p}{\partial w_{NE_{h1},NE_{h2}}^{h1h2}} \right) \left(\frac{\partial E_{tr_s}^p}{\partial b_1^{h2}} \; \frac{\partial E_{tr_s}^p}{\partial b_2^{h2}} \; \cdots \frac{\partial E_{tr_s}^p}{\partial b_{NE_{h2}}^{h2}} \right) \\
\left(\frac{\partial E_{tr_s}^p}{\partial w_{1,1}^{h2h3}} \; \frac{\partial E_{tr_s}^p}{\partial w_{2,1}^{h2h3}} \; \cdots \; \frac{\partial E_{tr_s}^p}{\partial w_{NE_{h2},1}^{h2h3}} \right) \cdots \left(\frac{\partial E_{tr_s}^p}{\partial w_{1,NE_{h3}}^{h2h3}} \; \cdots \; \frac{\partial E_{tr_s}^p}{\partial w_{NE_{h2},NE_{h3}}^{h2h3}} \right) \left(\frac{\partial E_{tr_s}^p}{\partial b_1^{h3}} \; \frac{\partial E_{tr_s}^p}{\partial b_2^{h3}} \; \cdots \frac{\partial E_{tr_s}^p}{\partial b_{NE_{h3}}^{h3}} \right) \\
\left(\frac{\partial E_{tr_s}^p}{\partial w_{1,1}^{h3o}} \; \frac{\partial E_{tr_s}^p}{\partial w_{2,1}^{h3o}} \; \cdots \; \frac{\partial E_{tr_s}^p}{\partial w_{NE_{h3},1}^{h3o}} \right) \left(\frac{\partial E_{tr_s}^p}{\partial b_1^o} \right)
\end{array} \right]$$

Table A1. Catalyst library – Representative data

Ni %	SBA-15 %	SiO_2	Gamma Alumina %	Alpha Alumina %	B %	Mo %	Zr %	Ce %	Temp (K)	Pressure (kPa)	S:C	Time on stream (min)	Catalyst Reduction Temperature (K)	Conversion %	Remarks
5	95	0	0	0	0	0	0	0	973	101.295	2.00	44400	1023	74.50	Wan et al,
10	90	0	0	0	0	0	0	0	1023	101.295	2.00	44400	1023	97.50	2007
15	85	0	0	0	0	0	0	0	1073	101.295	2.00	44400	1023	98.50	
10	80	0	0	0	0	0	10	0	1123	101.295	2.00	44400	1023	99.90	
10	80	0	0	0	0	0	5	5	1123	101.295	2.00	44400	1023	99.90	
10	80	0	0	0	0	0	0	10	1123	101.295	2.00	44400	1023	99.90	
60.2	0	0	39.8	0	0	0	0	0	973	101.295	4.00	105	1073	92.00	Maluf et al,
60.6	0	0	39.35	0	0	0.05	0	0	973	101.295	4.00	210	1073	94.00	2009
60.9	0	0	38.6	0	0	0.5	0	0	973	101.295	4.00	105	1073	83.00	
60.9	0	0	38.6	0	0	0.5	0	0	973	101.295	4.00	1615	1073	74.00	
59.7	0	0	39.3	0	0	1	0	0	973	101.295	4.00	689	1073	68.00	
60	0	0	38	0	0	2	0	0	973	101.295	4.00	365	1073	72.00	
60.2	0	0	39.8	0	0	0	0	0	973	101.295	2.00	17	1073	82.00	
60.6	0	0	39.35	0	0	0.05	0	0	973	101.295	2.00	59	1073	72.50	
60.6	0	0	39.35	0	0	0.05	0	0	973	101.295	2.00	500	1073	69.00	
59.7	0	0	39.3	0	0	1	0	0	973	101.295	2.00	137	1073	68.50	
60	0	0	38	0	0	2	0	0	973	101.295	2.00	213	1073	69.50	
60	0	0	38	0	0	2	0	0	973	101.295	2.00	387	1073	69.00	
20	0	80	0	0	0	0	0	0	773	101.295	2.00	30	773	21.80	Matsumura
20	0	0	80	0	0	0	0	0	773	101.295	2.00	30	773	0.00	et al, 2004
20	0	0	0	0	0	0	80	0	773	101.295	2.00	30	773	14.10	
5	0	0	0	0	0	0	95	0	773	101.295	2.00	240	773	21.30	
15	0	0	85	0	0	0	0	0	1073	101.295	1.00	115	1073	52.03	Xu et al,
15	0	0	85	0	0	0	0	0	1073	101.295	1.00	579	1073	44.47	2009
15	0	0	84.5	0	0.5	0	0	0	1073	101.295	1.00	93	1073	55.63	
15	0	0	84	0	1	0	0	0	1073	101.295	1.00	290	1073	58.24	
16	0	0	0	84	0	0	0	0	823	121.554	3.00	30	873	35.99	Hou et al,
16	0	0	0	84	0	0	0	0	823	121.554	3.00	60	873	30.48	2001
6.8	0	0	0	93.2	0	0	0	0	847	1002.8205	3.00	330	1023	6.49	
6.8	0	0	0	93.2	0	0	0	0	847	1002.8205	3.00	17259	1023	5.06	
6.8	0	0	0	93.2	0	0	0	0	872	1002.8205	3.00	1513	1023	5.03	Numaguchi
6.8	0	0	0	93.2	0	0	0	0	1010	1002.8205	3.00	248	1023	16.27	et al, 1995
6.8	0	0	0	93.2	0	0	0	0	1010	1002.8205	3.00	16358	1023	20.51	
6.8	0	0	0	93.2	0	0	0	0	1018	1002.8205	3.00	322	1023	16.59	
6.8	0	0	0	93.2	0	0	0	0	943	1002.8205	3.30	3464	1023	15.44	

*38 cases selected from a larger list of 200 data cases.

where, for instance the elements of the Jacobian are calculated as given below

$$\frac{\partial E^p_{tr_s}}{\partial w^{Ih1}_{n,m}} = \delta^{h1}_{m_s} y^I_n \qquad \frac{\partial E^p_{tr_s}}{\partial b^{h1}_m} = \delta^{h1}_{m_s}$$

where

n = 1, 2, …, NE_I represents the number of neurons in the input layer (I)

m = 1, 2, …, NE_{h1} represents the number of neurons in the first hidden layer (h1)

q = 1, 2, …, NE_{h2} denotes the number of neurons in the second hidden layer (h2)

r = 1, 2, …, NE_{h3} represents the number of neurons in the third hidden layer (h3)

and 'o' denotes the single output neuron present in the output layer

19. Increment p by one (i.e., p = p + 1).

20. If p > 1000, then restore all the weights and biases to weights(minimum) and biases (minimum), respectively and stop training. Go to step 28.

21. With the updated weights and biases, entire training data set is presented to the network again and the steps 2 to 14 are repeated and SSE^p_{tr} is calculated.

22. Steps 2 to 8 are repeated for all the data cases in validation set. The (p)th iteration mean squared error $MSE_{valid}(p)$ for validation set is calculated as given in Eq. A15 and stored in the memory.

23. Steps 2 to 8 are repeated for all the data cases in testing data set. The (p)th iteration mean squared error $MSE_{test}(p)$ for testing set is calculated as given in Eq. A16.

24. If $MSE^p_{valid} \geq [MSE_{valid}(minimum)]$ for 10th consecutive iteration than the training is stopped. This technique is called 'early stopping method' and by implementing this method, over fitting of the model can be avoided. Go to step 28.

25. If the $MSE^p_{valid} < [MSE_{valid}(minimum)]$ then value and the corresponding weights and biases replaces the values for $MSE_{valid}(minimum)$, weights(minimum) and bias(minimum), respectively in the memory.

26. If the $SSE^p_{tr} > SSE^{p-1}_{tr}$, then μ is multiplied by 10. If the $SSE^p_{tr} < SSE^{p-1}_{tr}$, then μ is divided by 10.

27. Go to step 18.

28. End

Representative data from the catalyst library are given in Table A1.

Manuscript received Jan. 16, 2011, and final revision received July 22, 2011.