Multilayer perceptron modeling for UASB Reactor treating tannery effluent
Parthiban.R1, Latha Parthiban2, Porselvam.S3, Ravindranath.E3
1- Department of Chemical Engineering, Sri Venkateswara College of Engineering, Sriperumbudur, India
2 - Department of Computer Science & Engineering, SSN College of Engineering, Kalavakkam, Chennai, India
3 - Scientist, Environmental Technology Division, Central Leather Research Institute, Chennai, India
rparthi@yahoo.com
doi:10.6088/ijes.00202030035

ABSTRACT
In this paper, the use of multilayer perceptron neural network for modeling is investigated for a bench scale system of up-flow anaerobic sludge blanket (UASB) reactor. The system study is the anaerobic digestion of synthetic wastewater derived from the starch processing industries. The experiment is carried out in a bench scale Up-flow Anaerobic Sludge Blanket reactor. It is proven that multilayer perceptron modeling has great adaptability to the variations of system configuration and operation condition. Multilayer perceptron neural network is trained with the experimental values obtained. The output parameters predicted for respective inputs have been found to be very much closer to the corresponding experimental ones and the model was validated by replicative testing.

Keywords: Artificial Neural Network, Multilayer Perceptron, UASB, Biomethanization.

1. Introduction
Artificial Neural Networks (ANNs) (Amari 1998, Duch 2007, Fausett 1994, Grossberg 1988) is an important research area in today’s Artificial Intelligence (AI) studies as they can tackle problems easily as they are equipped with a remarkable learning capability such that a desired input-output mapping can be discovered through learning by examples. ANNs can exhibit cognitive behaviors such as association, categorization, generalization, classification, feature extraction and optimization (Du and Samy 2008). These behaviors fall under the general categories of searching, representation and learning which are closely related to the associative property and self-organizing capability of the brain. Hagan M.T. and Menhaj M.B. (1994) used Marquardt algorithm for training feed forward networks. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons, which are true for neural networks as well.

Holubar et al (2002) applied neural networks based on the feed-forward back-propagation algorithm to model and control methane production in anaerobic digesters. Chen et al (2001) proposed a new approach for modeling industrial Waste Water Treatment Plants (WWTP). The approach employed a three-stage analysis that integrates fuzzy logic, genetic algorithms, and neural networks. Moreno-Alfonso and Redondo (2001) proposed an intelligent...
wastewater treatment concept, aided by two sets of neural networks, with the aim of controlling the plant in terms of previously selected parameters. The first neural network is used for reasoned (regular) control of the plant. The other neural network is termed instinctive, and is dedicated to the monitoring of the critical parameters. Zhang and Stanley (1999) studied the capacity of neural networks for process control in water treatment processes. Tay and Zhang (1999) integrated fuzzy systems and neural networks in modeling the complex process of anaerobic biological treatment of wastewater. They illustrated the power of the technique in two case studies of UASB and AFBR. Delgrange et al. (1998) applied neural networks to model the ultra-filtration membranes used in drinking water treatment. They experimented with a number of different network structures and architectures to compute the pressure at the end of filtration cycle and after the next backwash.

The paper is organized as follows: In Section II, we present the materials and methods of our experimental setup. Section III will brief about the modeling using MLP and its algorithm. In Section IV, we describe our results and discussions and Section V describes our conclusions.

2. Materials and Method

To study the biomethanization of tannery effluent, UASB reactor was fabricated. The characteristics of the effluent are presented in Table 1. The process flow diagram for bench scale studies carried out in UASB reactor is shown in Figure 1.

![Figure 1: Process flow Diagram of UASB reactor](image_url)

The top section, without temperature control, is provided with Gas- Liquid- Solid (GLS) phase separator. A magnetic joint is connected to the tube of the separator for a gas tight connection between the stirrer and the stirrer motor. The reactor is provided with a stirrer composed of motor – magnetic joint, glass connector, stainless steel axis, stirrer blades. The
The volume of reactor is five-litre. The inlets are the ends of a stainless pipe with a diameter of 10mm. The feed was fed into the reactor using a peristaltic pump. The treated effluent was recycled from the top through another peristaltic pump with adjustable speed controller. Biogas was passed through soda lime pellets and then through wet gas flow meter to measure the methane generated. Temperature of the reactor was maintained at 30°C±1 by water circulation through thermostat. The feed and the outlet characteristics are analyzed as per the ‘Standard Methods for the examination of water and wastewater’

3. Modeling using Multilayer Perceptron Neural Network

3.1 Development of Multilayer Perceptron model

Multilayered Perceptron is a supervised neural network which consists of multiple layers of Processing Elements (PE) connected in a feed forward fashion. It consists of three types of layers: input, hidden, and output. It has a one-directional flow of information, generally from the input layer, through hidden layer, and then to the output layer, which then provides the response of the network to the input stimuli. In this type of network, there are generally three distinct types of neurons organized in layers. The input layer contains as many neurons as the number of input variables. The hidden neurons, which are contained in one or more hidden layers, process the information and encode the knowledge within the network. The hidden layer receives, processes, and passes the input data, to the output layer. The selection of the number of hidden layers and the number of neurons within each affects the accuracy and performance of the network. The output layer contains the target output vector. The architecture of multilayer perceptron is shown in Figure 2.

Table 1 Characteristic of UASB feed details on Tannery effluent

| S.No | Parameter               | Minimum | Maximum  | Mean value |
|------|-------------------------|---------|----------|------------|
| 1    | pH                      | 7.2     | 8.2      | 7.4        |
| 2    | COD (Total) (mg/L)      | 3500    | 14364    | 10980      |
| 3    | Total Solids (TS) (mg/L)| 13200   | 22515    | 15200      |
| 4    | Suspended Solids (SS) (mg/L) | 2550 | 4270 | 3379 |

A weight coefficient is associated with each of the connections between any two neurons inside the network. Information processing at the neuron level is done by an “activation function” that controls the output of each one. ANNs (Hinton G.E. and Salakhutdinov R.R. (2006), train through adaptation of their connection weights based on examples provided in a training set. The training is performed iteratively until the error between the computed and the real output over all training patterns is minimized. Output errors are calculated by comparing the desired output with the actual output. Therefore, it is possible to calculate an error function that is used to propagate the error back to the hidden layer and to the input
A layer in order to modify the weights. This iterative procedure is carried out until the error at the output layer is reduced to a pre-specified minimum or for a pre-specified number of epochs. A plot is made between the epoch and MSE. Error correction learning works in the following way: From the system response at PE \( i \) at iteration \( n \), \( y_{i(n)} \) and the desired response \( d_{i(n)} \) for a given input pattern an instantaneous error \( e_{i(n)} \) is defined by

\[
e_{i(n)} = d_{i(n)} - y_{i(n)}
\]  

(1)

![Figure 2 Architecture of Multilayer Perceptron neural network](image)

Using the theory of gradient descent learning, each weight in the network can be adapted by correcting the present value of the weight with a term that is proportional to the present input and error at the weight, i.e.

\[
w_{ij}(n+1) = w_{ij} + \eta \delta_i(n)x_j(n)
\]  

(2)

The local error \( \delta_{i(n)} \) can be directly computed from \( e_{i(n)} \) at the output PE or can be computed as a weighted sum of errors at the internal PEs. The constant \( \eta \) is called the step size. The backpropagation algorithm is most commonly used for training MLP and is based on minimizing the sum of squared errors between the desired and actual outputs (Judd 1990; Kung 1993; Lau 1992). If the number of input attributes is more, it can be reduced using ANN without loss of prediction accuracy.

### 3.2 Multilayer Perceptron Algorithm

The algorithm for the multilayer perception is shown below. It requires the units to have thresholding non-linear functions that are continuously differentiable, i.e. smooth everywhere. A sigmoid function

\[
f(\text{net}) = \frac{1}{1+e^{-k \text{net}}}
\]  

(3)

is used, since it has a simple derivative. All training and testing data were normalized.
i) Initialize weights and thresholds. Set all weights and thresholds to small random variable.

ii) Present input and desired output. Present input
\[ X_p = x_0, x_1, x_2, \ldots, x_{n-1} \tag{4} \]
and target output
\[ T_p = t_0, t_1, \ldots, t_{m-1} \tag{5} \]
where \( n \) is the number of input nodes and \( m \) is the number of output nodes. Set \( w_0 \) to be \(-\theta\), the bias, and \( x_0 \) to be always 1. For pattern association, \( X_p \) and \( T_p \) represent the patterns to be associated. For classification, \( T_p \) is set to zero except for one element set to 1 that corresponds to the class the \( X_p \) is in.

iii) Calculate actual output
Each layer calculates
\[ y_{pj} = f \left[ \sum_{i=0}^{n-1} w_{ij} x_i \right] \tag{6} \]
and passes that as input to the next layer. The final layer outputs values \( o_{pj} \) and passes that as input to the next layer.

iv) Adapt weights. Start from the control layer, and work backwards.
\[ w_{ij}(t+1) = w_{ij}(t) + \eta \delta_{pj} o_{pj} \tag{7} \]
where \( w_{ij}(t) \) represents the weights from node \( i \) to node \( j \) at time, \( \eta \) is a gain term, and \( \delta_{pj} \) is an error term for pattern \( p \) on node \( j \).

For output units
\[ \delta_{pj} = k \ o_{pj} (1-o_{pj}) \ (t_{pj} - o_{pj}) \tag{8} \]
For hidden units
\[ \delta_{pj} = k \ o_{pj} (1-o_{pj}) \sum_{k} \delta_{pk} w_{jk} \tag{9} \]
where the sum is over the \( k \) nodes in the layer above node \( j \).

The stopping condition may be weight change, number of epochs, etc

4. Results and Discussions

In the course of the system training, the data were taken during the steady state operation of the reactor (when the efficiency of the reactor was 90 – 92%). Each of them was defined by four inputs (flow rate, hydraulic retention time, COD influent and VFA influent) and three outputs (COD effluent, VFA effluent and biogas generation).

Trained ANNs through adaptation of their connection weights were based on examples provided in a training set. The training is performed iteratively until the error between the computed and the real output over all training patterns is minimized. Output errors are calculated by comparing the desired output with the actual output. Therefore, it is possible to calculate an error function that is used to propagate the error back to the hidden layer and to the input layer in order to modify the weights. This iterative procedure is carried out until the error at the output layer is reduced to a pre specified minimum or for a pre specified number of epochs. A plot of epoch and MSE is presented in Figure 3 for MLP.
Multilayer perceptron modeling for UASB Reactor treating tannery effluent

Parthiban.R, Latha Parthiban, Porselvam.S, Ravindranath.E
International Journal of Environmental Sciences Volume 2 No.3, 2012

Figure 3: Learning curve of the MLP neural network

Figure 4: Output vs desired plot for the MLP model

Actual validation of an already trained ANN requires testing the network performance on an exclusive set of data, called testing data, which is composed of data that was never presented to the network before. Cross validation computes the error in a test set at the same time that the network is being trained with the training set. If the error obtained in both training and testing phases is satisfactory, the NN is considered adequately developed and thus can be used for practical applications.

The output versus the measured data (desired) is shown in Figure 4 for MLP.
5. Conclusions

In this paper, we have applied MLP to model the parameters for the treatment of tannery effluent using UASB reactor. This is done by employing MLP to identify the nonlinear relationship between the various experimental parameters. We have validated our technique by replicative testing and had testing error of 0.00123 for biogas generation, 0.0208 for VFA effluent and 0.006 for COD effluent which is very much encouraging for further research in this area. Further no such work on MLP modeling in the area of waste water treatment has been reported in the literatures. Our future work will concentrate on modeling with other types of ANNs like SOM and SVM to analyze and compare their performances.

6. References

1. Amari S. (1998), “Natural gradient works efficiently in learning”, *Neural computation*, 10(2), pp 251-276.

2. Chen W.C., Chang N.B. and Shieh W.K. (2001), “Advanced hybrid fuzzy-neural controller for industrial wastewater treatment”, *Journal of Environment Engineering*, American Society of Civil Engineers, 127(11), pp 1048-1059

3. Delgrange N., Cabassud C., Cabassud M., Durand-Bourlier L. and Laine J.M. (1998), “Neural networks for prediction of ultrafiltration transmembrane pressure: application to drinking water production”, *Journal of Membrane Science*, 150(1), pp 111-123.

4. Du K.L. and Swamy M.N.S. (2008), “Neural networks in a soft computing framework”, Springer (India) Private Limited.

5. Duch W. (2007), “What is computational intelligence and Where it is going”, in: W.Duch and J.Mandziuk (Eds.), *Challenges for Computational Intelligence*, Springer Verlag, Heidelberg, pp. 1 -13,

6. Duda R., Hart P. and Stork D. (2001), “Pattern Classification”, II Edition. John Wiley and Sons, Inc.

7. Fausett L. (1994), “Fundamentals of Neural Networks – Architectures, Algorithms, and Applications”, Prentice-Hall, Inc., New Jersey.

8. Grossberg S.A. (1988), “Neural networks and Natural Intelligence”, MIT Press, Cambridge, MA.

9. Hagan M.T. and Menhaj M.B. (1994), “Training feed forward networks with the Marquardt algorithm”, *IEEE Transactions on Neural Networks*, 5(6) pp 989–993.

10. Ham F.M. and Kostanic I. (2001), “Principles of Neurocomputing for Science and Engineering”, McGraw Hill, New York.

11. Hinton G.E. and Salakhutdinov R.R. (2006), “Reducing the Dimensionality of Data with Neural Networks- Materials and Methods”, Science, 313(5786), pp 504 – 507.
12. Holubar P., Zani L., Hager M., Froschl W., Radak Z. and Braun R. (2002), “Advanced controlling of anaerobic digestion by means of hierarchical neural networks”, Water Research, 36(10), pp 2582-2588.

13. Judd J. S. (1990), “Neural network design and the complexity of learning”, MIT Press.

14. Kung S.Y. (1993), “Digital Neural Networks”, Prentice Hall, U.S.A.

15. Lau C. (1992), “Neural networks: Theoretical foundations and analysis”, IEEE Press.

16. Moreno-Alfonso N. and Redondo C.F. (2001), “Intelligent waste-water treatment with neural-networks”, Water Policy, 3, pp 267-271.

17. Tay J.H. and Zhang X. (1999), “Neural fuzzy modeling of anaerobic biological wastewater treatment systems”, Journal of Environment Engineering, American Society of Civil Engineers, 125(12), pp 1149-1159.

18. Zhang Q. and Stanley S.J. (1999), “Real-time water treatment process control with artificial neural networks”, Journal of Environment Engineering, American Society of Civil Engineers, 125(2), pp 153-160.