NEURAL NETWORK APPLICATIONS IN POLYMERIZATION PROCESSES

F. A. N. Fernandes[1*] and L. M.F. Lona[2]

[1]Departamento de Engenharia Química, Universidade Federal do Ceará,
Campus do Pici, Bloco 709, 60455-760, Fortaleza - CE, Brazil.
E-mail: fabiano@efftech.eng.br
[2]Departamento de Processos Químicos, Faculdade de Engenharia Química,
Universidade Estadual de Campinas, C. P. 6066, 13083-970 Campinas - SP, Brazil.
E-mail: liliane@feq.unicamp.br

Abstract - Neural networks currently play a major role in the modeling, control and optimization of polymerization processes and in polymer resin development. This paper is a brief tutorial on simple and practical procedures that can help in selecting and training neural networks and addresses complex cases where the application of neural networks has been successful in the field of polymerization.
Keywords: Neural network; Polymerization; Simulation.

INTRODUCTION

The use of neural networks (NNs) has become increasingly popular for applications where the mechanistic description of the interdependence of dependent and independent variables is either unknown or very complex. They are now the most popular artificial learning tool with applications in areas such as pattern recognition, classification, process control, optimization (Hanai et al., 2003; Krothapally & Palanki, 1997; Nascimento et al., 2000; Syu & Tsao, 1993; Tian et al., 2001; Tsen et al., 1996; Zhang, 1999).

In the past decade many peer-reviewed articles showing good results on the application of NNs were published in the literature. But several studies with NNs failed due to the poor predictions outputted by the NN. These failures resulted in some criticism of the ability of NNs to deal with some kinds of processes. In part, the criticism is founded, since although NNs have been know for some time, they are still in the early stages of development of their underlying theory and many improvements in their structure can be made, but many applications made, failed because researchers did not try to use more than one hidden layer in the NN topology or present the NN with enough data for training. In this paper, some well-established training procedures are presented as a brief tutorial of recommend procedures and some applications are shown for more complex cases in the field of polymerization.

NEURAL NETWORKS

Most papers on the use of NNs apply a multilayered, feed-forward, fully connected network of perceptions. Reasons for the use of this kind of NN are the simplicity of its theory, ease of programming and good results and because this NN is a universal function in the sense that if topology of the network is allowed to vary freely it can take the shape of any broken curve. Figure 1 shows the scheme of this kind of NN.

In general, the network consists of processing neurons and information flow channels between the neurons, usually called "interconnects". Each

---

*To whom correspondence should be addressed

processing neuron calculates the weighted sum of all interconnected signals from the previous layer plus a bias term and then generates an output through its activation transfer function. The transfer functions associating individual nodes have typically a sigmoid shape such as

$$y = \frac{1}{\left(1 + e^{a.x}\right)} \qquad (1)$$

Other transfer functions, such as hyperbolic tangent functions, can also be applied. The adjustment of the NN function to experimental data (learning process or training) is based on a non-linear regression procedure (Fraser, 2000). Training is done by assigning random weights to each neuron,

evaluating the output of the network and calculating the error between the output of the network and the known results by means of an error or objective function. If the error is large, the weights are adjusted and the process goes back to evaluate the output of the network. This cycle is repeated till the error is small or a stop criterion is satisfied. More information regarding basic NN theory and training procedure can be found in Haykin (1998) and White (1992).

During the last few years, some procedures for using NNs have become well established and are used and recommended by several researchers (scaled inputs and data concentration) and some procedures are recommended but are still not used by many researchers (cross-validation and early stop criterion).
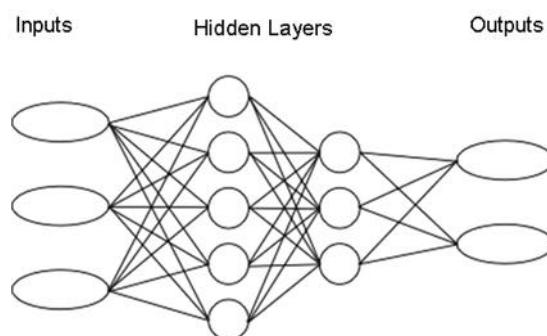

Figure 1: Basic neural network scheme.

Scaled Inputs

Independent and dependent variables should be scaled within the same range or the same variance and shifted to the general region of NN initial conditions. Scaling is frequently done between 0 and 1, but a good training technique is to scale the independent variables between 0.1 and 0.9. Scaling can be done applying the formula

$$Y_{norm} = 0.8 \left(\frac{Y - Y_{min}}{Y_{max} - Y_{min}} \right) + 0.1 \qquad (2)$$

Data Concentration

For overall approximation, the training data should uniformly cover the entire design space (region between the lower and upper limits of each independent variable). If the training data has too many points concentrated in a given region and some sparse data in the rest of the design space, the NN will tend to overfit the data and the output of the NN will tend towards the region where more data were

available at training. Therefore special attention should be paid to having a uniform distribution of data throughout the design space.

Simple Cross-Validation

When training NNs, the prediction error is evaluated for each iteration. If a NN with too many neurons is used, it will allow an excess of degrees of freedom that can cause overfitting of the data, i.e., the NN will train only to predict the training data set, losing its ability to correlate other data sets. A cross-validation set of data can be separated and used only to check how good the fit of the NN is, based on the sum of squared prediction errors. The optimal degree of training can be obtained when the sum of the training plus cross-validation errors are at a minimum (Figure 2). Some attention should be paid so as not to stop at the first minimum point. Training should be allowed to proceed further to check whether or not it is a point of local minimum, since a local minimum can be found. More information on advanced cross-validation can be found in Wold (1978).
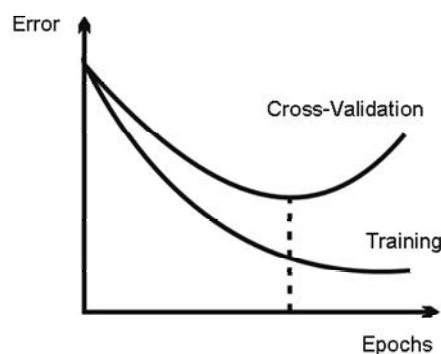
Figure 2: Profiles for training and cross-validation errors.

Stop Criteria

There are several choices in deciding when to stop training the NN. Training can be stopped when to a predefined number of epochs (iterations) are reached, when the error function becomes small, when the gradient of the error function becomes small, or when the cross-validation error becomes small. The cross-validation criterion is highly recommended, since it prevents the NN from overfiting the data and because no minimum error value needs to be specified (which can itself be hard to establish). This procedure is sometimes called early stop by cross-validation.

ACHIEVEMENTS AND CHALLENGES WITH NEURAL NETWORKS

Currently, NNs are basically used to mimic mathematical models, classify data or estimate some product properties or as a substitute for an unknown model, which uses simple NNs with one hidden layer and a few neurons in the hidden layer. Use in decision making, complex optimization and inverse modeling in the field of polymerization is not frequent and has been avoided by many researchers simply because these applications need NN topologies or training procedures that do not comply with the ground rules established for parameter estimation techniques, specially regarding the number of weight of the NN and the number of data available to train the NN.

When treating NNs as a class of parameter estimation methods, the same ground rules that are adopted with traditional parameter estimation techniques are also applied to NNs. A ground rule such as the need for more data points than the number of parameters to be estimated, when applied to neural networks, implies a having more data sets than the number of weights used in the NN (which is not necessary). These ground rules have limited the advances that we should be getting from NNs.

When NNs were first idealized, they were taught to behave as universal approximators capable of mimicking the human mind, learning to correlate inputs and outputs. Hornik, Stinchcombe and White (1989) proved that standard multilayer feedforward networks are capable of approximating any measurable function to any desired degree of accuracy, establishing that these NNs are universal approximators. Therefore the lack of success in many applications arises from inadequate learning, insufficient numbers of hidden units or the lack of a deterministic relationship between input and target. Advances in NN application and research will occur if we can break with the current paradigm that NNs should be treated as any other parameter estimation method. In the following sections some nontraditional procedures and NN examples are shown and discussed.

Training Data

Success in obtaining a reliable and robust network depends sheavily on the choice of process variables involved, as well as the available data set and the domain used for training purposes. A problem with NN-based models is that they can lack generalization capability if not properly trained and if the data available for training is insufficient. This problem can be overcome by carefully selecting the range of data points and the form of selecting and presenting the data to the NN and through hybrid modeling, where a simplified mechanistic model is supplemented by a neural network model, and through the combination of multiple NNs.

Range of Data Points

Neural networks do not always predict well near the borders of their training range (what we call the shadow zone). Extending the training range so that the region of interest falls within 85 to 95% of the total training range can minimize the border problem. For example, if the inputs are the concentrations of A and B which range from 0.1 to 0.5 (A) and from 2.0 to 10.0 (B), then the training range should, if possible, be increased to range from 0.08 to 0.52 (A) and from 1.55 to 10.45 (Figure 3).

Random Data Points

Data used to train the NN can be gathered or selected using a full factorial design, using random points or a mixture of factorially designed points and random points. Experience has shown that training the NNs with random points lowers the error of the predictions with the training and testing data sets, where the term "training points" refers to the data set used to train the NN, while "testing points" refers to the data set used to test the prediction capabilities of the NNs (testing points differ from training points and are presented to the NN only at the testing stage, never at the training stage).

In a study to determine the operational conditions of vinyl acetate emulsion polymerization based on the information on the desired polymer characteristics (inverse modeling), we have addressed the problem of factorially designed data against random data. When randomly generated data were used, the mean prediction error for all variables decreased and the results were especially impressive for very sensitive variables such as initiator concentration (Figure 4).

An explanation for the better results obtained for randomly selected data points than for points obtained by factorial design is that when points from a star factorial design for two variables are fed to the NN, the NN will work with nine inputs, but only five different input values (Figure 5a). On the other hand, if nine randomly selected data points are fed to the NN, the NN will work with nine inputs and up to nine different input values (Figure 5b). Using evenly spaced points, the same inputs will generate only three different input values, reducing the number of different input values (Figure 5c).
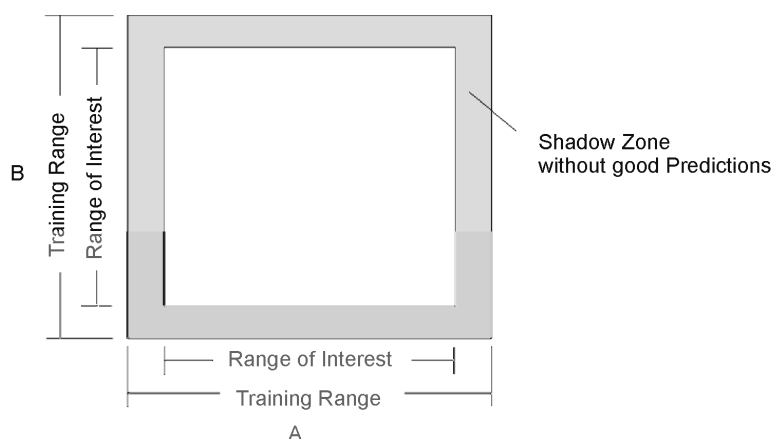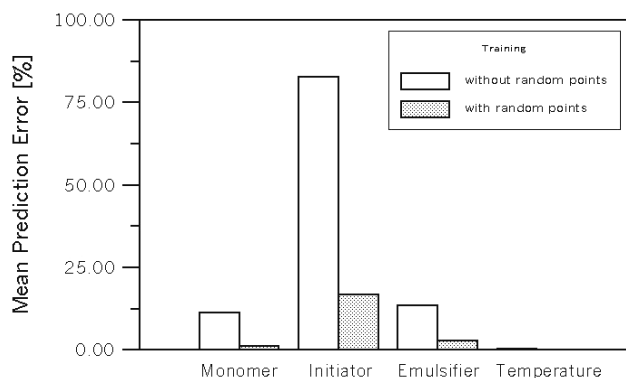

Figure 3: Training range.

Figure 4: Mean prediction errors outputted by a NN when data were selected using star factorial design and randomly generated points (5-20-25-4 NN topology; training points = 298; testing points = 96).
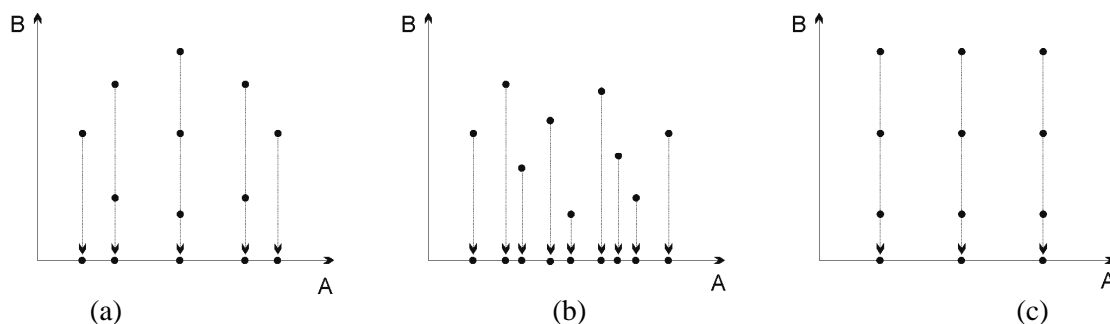
Figure 5: Selection of data points using star factorial design (a), randomly generated points (b) and evenly spaced points (c).

The effect of the random data points is that they can cover the region of variables better and more evenly, therefore providing better information about the response in this region. An important consideration is that not only should the region of variables be well covered, but also each variable should have a distributed coverage, with as many different values as possible. Evenly spaced points will have the same response as random points but more data points will be required to cover the entire region of variables and to provide enough different values for each variable than for randomly selected data points.

Data Gathering

Design and conduction of a large number of experiments to generate data for NN training is expensive and time consuming and may not be acceptable either in a production environment or to experiment with NNs capabilities. Thus the use of a mathematical model to generate the data for NN training is almost essential, unless there is a large amount of data for the process. The use of a mix of simulation data and experimental results is welcome, and experimental results should be used when testing and cross-validating the NN so as to ensure that the simulation data and also the NN predictions are satisfactory.

A typical question that arises regarding the need for data from a mathematical model to train the NN, is "why use a NN if I already have a mathematical model?" A NN outputs an answer much faster than a mathematical model, especially if the mathematical model is complex. This enhanced speed can be very welcome for optimization problems (like in inverse modeling) and for process control where the response to a disturbance must be almost instantaneous.

NN Topology

Prior to training and using a neural network, the best topology and the best way to train must be found. First, potentially good topologies must be identified. Nevertheless, no good theory or rule accompanies the NN topology that should be used and trial-and-error is still required. This is done by testing several topologies and comparing the prediction errors. Smaller errors indicate potentially good topologies, i.e., neural network topologies with chances to train well and to output good results.

Regarding the best topology, it may be dangerous simply to assume that a single hidden layer will naturally provide adequate approximations. For the case of a single hidden layer, it should be noted that a given degree of accuracy is attained only in the limiting case as the number of nodes becomes infinitely large. For most complex cases, additional hidden layers are required, not only from the point of view of a good fit but also additional layers provide an improved capacity to generalize. With a finite set of data, it is possible to reduce the number of network weights by adding layers (Morgan et al., 1999; Curry et al., 2000, 2002). Still, as a future challenge in the NN field, there is a need for further theoretical research on the impact of the number of nodes and hidden layers and how to optimize the relationships between them.

Aside from the theoretical considerations, we offer some practical tips for searching for potentially good topologies depending on three different classes of NNs (Figure 6). We have proposed these classes so as to differentiate the NNs according to the ratio of number of input variables to number of output variables. Class I refers to the NN that has more input than output variables, while Class II refers to NNs with an equal number of inputs and outputs and Class III refers to the NN which has more output than input variables.
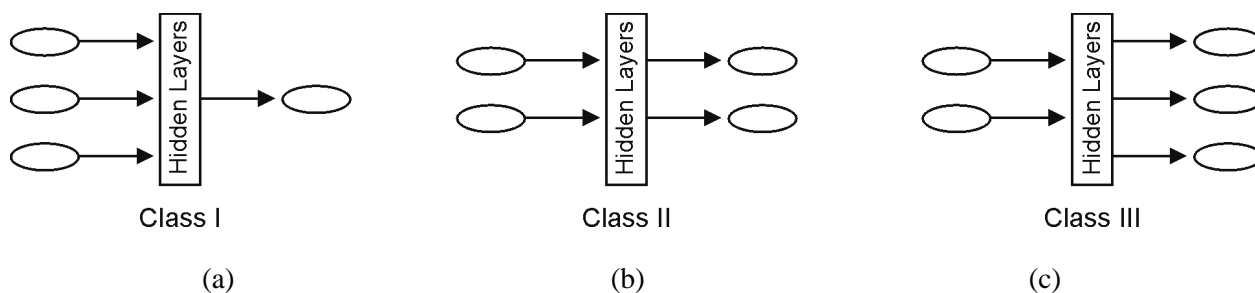
Figure 6: Classes of NNs

For Class I NNs (more inputs than outputs), one hidden layer is enough in most cases and, according to Tamura and Tateishi (1997), if *N-1* neurons are used in the hidden layer (where *N* is the number of inputs), the NN will give an exact prediction. This recommendation works well when the system has a small number of inputs and the correlation between the data points (inputs and outputs) is not very complex; otherwise their recommendation will not always work and we recommend the use of 8 to 20 neurons in the hidden layer for better precision and shorter training time. If the number of outputs is equal to or higher than four and they are not independent (one cannot be estimated without the others), then a second hidden layer might be needed for better predictions.

For Class II NNs (same number of inputs and outputs), one hidden layer is not always enough and a NN with two hidden layers is recommended in order to enhance its ability for generalization. If one hidden layer is used we recommend the use of 20 to 40 neurons in the hidden layer. If two hidden layers are used, we recommend the use of 13 to 20 neurons in the first hidden layer and from 18 to 25 neurons in the second hidden layer (five more neurons than what was used in the first hidden layer).

For Class III NNs (more outputs than inputs), two or three hidden layers are needed. If two hidden layers are used, we recommend the use of 10 to 20 neurons in the first hidden layer and from 15 to 25 neurons in the second hidden layer. If a third hidden layer is used, this layer should have the same number of neurons as the second layer.

The recommendations made for the number of hidden layers and neurons in each layer will provide the user with a NN that will function and will most probably made good predictions, but will not provide an optimized NN (which would require a statistical analysis of the neuron responses and elimination of excess neurons).

Stacked Neural Networks

To avoid the process of training several NNs and selecting the best one, stacked neural networks can be used (Figure 7). Instead of selecting a single best NN, a stacked network which combines a number of NNs can improve overall representation accuracy and robustness (Zhang et al., 1997; Zhang, 1999).

The overall output of the stacked NN is a weighted sum of the individual NN outputs:

$$Y = \sum w_i \cdot y_i \tag{3}$$

where Y is the stacked NN predictor, y_i is the *i*th NN predictor and w_i is the stacking weight for the *i*th NN.

Since each NN can behave differently in different regions of the training range (or space), the combination of results from two or more NNs can be more accurate, since a bad result from one NN can be compensated for by two good results from the other NNs. In Table 1, the benefit of a stacked NN is shown for an inverse modeling case, where the concentration of three initiators and the process temperature were searched in order to produce polystyrene with a given molecular weight and polydispersity (additional information on this case study is given in section 4.2). As shown in Table 1, the prediction error for a stacked neural network can be lower than the individual errors for each NN that have been combined.

This kind of NN can be of special interest to those who want to apply NNs but do not want to select the best NN. At any rate, a notion of potentially good NNs is needed or the predictions will fail. The example in Table 2 shows how the stacked NNs will not output good results when bad NNs are selected.
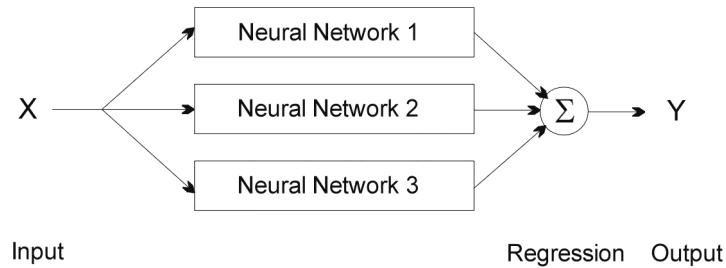
Figure 7: Stacked NN scheme.

Table 1: Example of a stacked NN.

NN	Initial Condition				Prediction Error			
	Vazo 52 [mol/L]	Vazo 64 [mol/L]	Vazo 88 [mol/L]	Temp [K]	Vazo 52 [%]	Vazo 64 [%]	Vazo 88 [%]	Temp [%]
4-20-25-4	0.090	0.060	0.025	376.3	4.62	8.63	7.22	0.39
4-15-20-20-4	0.084	0.054	0.029	378.5	2.27	1.95	6.25	0.18
4-20-25-25-4	0.088	0.055	0.028	377.2	1.65	0.16	4.25	0.17
Stacked	0.088	0.056	0.027	377.3	1.33	2.17	1.09	0.13
Known Data	0.086	0.055	0.027	377.8	-	-	-	-

Table 2: Example of a stacked NN.

NN	Initial Condition				Prediction Error			
	Vazo 52 [mol/L]	Vazo 64 [mol/L]	Vazo 88 [mol/L]	Temp [K]	Vazo 52 [%]	Vazo 64 [%]	Vazo 88 [%]	Temp [%]
4-10-15-4	0.096	0.053	0.024	377.4	11.45	3.41	10.10	0.11
4-15-20-4	0.090	0.057	0.019	377.1	3.93	4.67	28.60	0.21
4-20-25-4	0.090	0.060	0.025	376.3	4.62	8.63	7.22	0.39
Stacked	0.092	0.057	0.023	376.9	6.67	3.30	15.31	0.24
Known Data	0.086	0.055	0.027	377.8	-	-	-	-

Number of Data Samples

A sufficient number of data points should be used to guarantee good NN training. If the rules for parameter estimation are to be applied, the number of data points for training should be at least ten times the number of weights to be estimated, but this rule is extremely overestimated for neural networks. There is still no formula to estimate the number of data points required to train a NN, and the number can vary greatly depending on the complexity of the problem and the quality of the data, but many NNs have been trained successfully with smaller number of data points than number of weights. An optimization of the number of data sets that is really needed is still a challenge in the field of NNs.

A NN model is a set of computational rules associated with a network that tries to simulate the network of human neurons learning from experience. And like the human brain, NNs can learn to correlate data and generalize relationships with a small and limited set of data that is sufficient to learn the correlation and are insensitive to too many data points. This is the same thing that happens to us, humans. We learn the consequences of an action under different circumstances, by experiencing it or watching others. After a few experiences (data points) we can correlate the consequence of that action under any new circumstance. We do not need to experience it several billions of times (we have billions of neurons and therefore several billions of weights), as we are insensitive to too many data points. Unfortunately, the functionality of the NN cannot represent the complexity of the human brain, it cannot "think", but in a way it can "learn" from the data presented to the NN.

In some of our research, we addressed the inverse modeling of polymerization reactors, using Class II neural networks to optimize polymerization systems and Class III neural networks to estimate the operating conditions of a polymerization reactor based on the information on polymer properties needed for a given application. In a study on the selection of initiator mixtures for styrene

polymerization using NN, the errors of the individual variables as a function of the number of training points were examined. The trained NN should be able to select an initiator mixture and an operating temperature that are appropriate for producing polystyrene with a given molecular weight and polydispersity but it was constrained by the maximum amount of heat that could be produced by the reaction (the reactor cooling system had a maximum capacity for heat removal – additional information on this case study is presented in section 4.2).

The NN was presented with different numbers of data in the learning set (the testing set remained the same) and it was observed that 300 data points were sufficient to guarantee good predictions and above 300 data points the NN did not improve its predictions much (prediction errors remained at a constant level) (Figure 8).
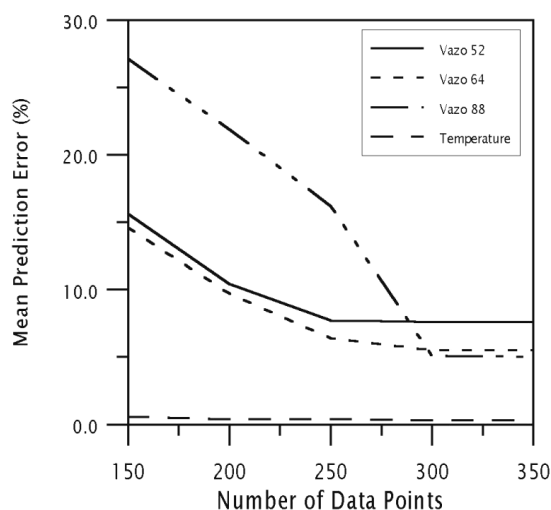

Figure 8: Prediction errors as a function of the number of data points in the training data set.

The influence of the size of the learning set in a neural network used in the inverse modeling of an emulsion reactor (vinyl acetate polymerization) was also studied. The NN consisted of two hidden layers with 20 and 25 neurons respectively in the first and second hidden layers. The network was provided with different numbers of data points in the learning set, and the results showed that 298 points were sufficient to guarantee good NN training and that fewer data points increased the prediction errors. Learning sets with more than 300 points did not enhance the training procedure, and therefore the quality of prediction was insensitive to a larger set.

In the examples cited, a 4-20-25-25-4 NN was trained successfully with 296 data points and a 5-20-25-4 NN was trained with 298 data points. If parameter estimation recommendations were followed, the number of data points should be at least ten times the number of weights. In these NNs, the number of weights are 1205 and 680 respectively for the 4-20-25-25-4 NN and the 5-20-25-4 NN, and the estimation of the weights would require 12050 and 6800 data points respectively, or more than 30 times the number of data points needed.

As a starting point for the number of data points required to train a NN, we recommend the use of 20 times the number of inputs x outputs. This number is generally greater than the number of points that made training insensitive to extra data. And the number of total hidden neurons should be about four times greater than the number of inputs x outputs displayed in many hidden layers, as recommended in section 3.2. Using these numbers, we always obtained prediction errors between 2 and 10%. Reducing the number of data points, the prediction errors increased.

Recurrent Neural Networks

In the on-line control of polymerization processes, application of recurrent NNs can be useful (Figure 9) (Tian, et al., 2001; Xiong & Zhang, 2005). Recurrent NNs are similar to a multilayered, feed-forward, fully connected network of perceptions, but one or more of the inputs (at time t) are the outputs of the NN at times $t-1$, $t-2$ and others. The lagged network outputs are fed back to the network input nodes as indicated by the back-shift operator **z**. In this way, dynamics are introduced into the network, and thus the network output depends not only on the network inputs, but also on the previous network outputs.
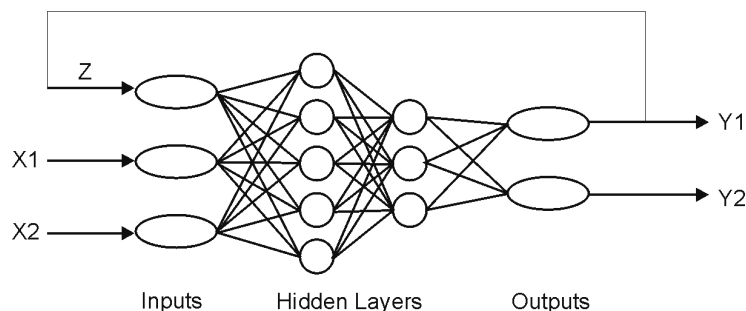
Figure 9: Recurrent NN scheme.

RECENT ACHIEVEMENTS IN THE FIELD OF POLYMERIZATION

Most applications of NNs in the field of polymer uses the Class I NN to predict end-user properties based on the molecular weight distribution of the polymer or the physical characteristic of the polymer (Al-Hiak et al., 2004; Ebube et al., 2000; Fujii et al., 2003; Hinchliffe et al., 2003; Huang & Liao, 2002; Kuroda & Kim, 2002; Simon & Fernandes, 2004; Sumpter & Noid, 1994; Sun et al., 1996; Zhang et al., 2003). This application is usually simple and does not require many data points to train the NN. Problems with multiple answers do not generally occur in this application; thus the use of these NNs is very direct and difficulties are not common.

In this section we present cases where Class II and Class III NNs are used with success in order to design, optimize and control polymerization processes.

Inverse Modeling

A novel use of NNs is their application in inverse modeling and reactor optimization. Inverse modeling is the name given to the search process whereby the results of a system are used to obtain its initial conditions. For polymerization reactors, this means that the initial operating conditions are obtained based on the product quality desired. Few papers have been published on this subject (Hanai et al., 2003), since these applications involve Class II and Class III NNs and are very complex to train, requiring two or three hidden layers and at least 15 neurons per hidden layer. The major difficulty in training NNs in these cases is that the problem may lead to multiple answers.

A mathematical model can easily predict the polymer properties from the inputs of reactor conditions, but the other way around (inverse modeling) is much more difficult and an optimization technique must be used. In the past few years we have applied inverse modeling to fluidized bed, emulsion and batch reactors with very good results when predicting the operating conditions of the reactor based on the product quality desired (Fernandes & Lona, 2002; Fernandes, 2002; Fernandes, et al., 2004).

A study to determine the operating conditions of gas-phase ethylene polymerization in a fluidized bed reactor, based on information on the desired polymer characteristics, focused on determining a NN structure, as shown in Figure 10, that could handle prediction of the operating conditions of the reactor, since a simple NN with one hidden layer failed to output good results. The reactor is very complex and at least six variables must be set for the operating conditions: gas feed rate (monomer), catalyst feed rate, gas superficial velocity, porosity, pressure and temperature. The number of variables that need to be specified increases if copolymerization is employed. In this case, gas feed rate for the monomers and the comonomer must be known and this study was carried out with an ethylene and 1-butene copolymer.

Data used in training were selected using a full 3^n factorial design, covering the whole range of operation of the reactor. Points for the factorial design were taken as the lowest, medium and highest values of the range of operation of each variable. Reactor pressure was set at 25 atm and was not used in the NN training. The ranges of the operating conditions used for each variable are presented in Table 3.
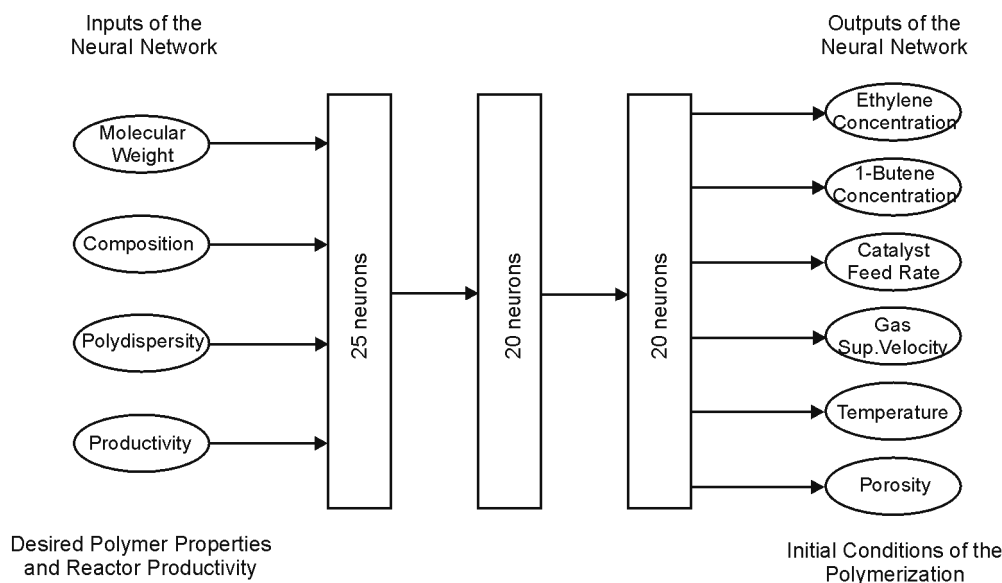
Figure 10: Class III NN used in the gas-phase ethylene polymerization inverse modeling problem (number of neurons shown refers to the best topology encountered).

Table 3: Ranges used for each operating condition.

Variable	Range	
	Minimum	Maximum
Ethylene Concentration [mol/L]	0.275	0.400
1-Butene Concentration [mol/L]	0.080	0.275
Feed Temperature [K]	310	350
Catalyst Feed Rate [g/s]	0.05	0.50
Superficial Velocity [x U_{mf}]	3	6
Porosity	0.40	0.60

Some data from the factorial design lay beyond the limit of physical capability of the reactor and thus were omitted. The final amount of data available for training was 176 points, of which 20% (35 points) were used as testing points. A smaller number of points (2^n factorial design) was tested, but the deviations between the NN predictions and the simulation data were greater than 10%, while a maximum of 2% deviation would be expected as a good prediction for this kind of application. Deviations were calculated as

$$e = \left| \frac{(\text{simulation data}) - (\text{NN prediction})}{(\text{simulation data})} \right| \cdot 100 \ [\%] \quad (4)$$

Several topologies from one to three hidden layers were tested, and the best results were obtained with a three hidden-layer NN with 25, 20 and 20 neurons, respectively, in the first, second and third hidden layers. This was the smallest network that had output predictions with less than a 2% deviation between NN prediction and simulation data. Tables 4 and 5 show

typical examples of the quality of the NN prediction for this inverse modeling problem. Standard error for all predictions (training and testing data) was 1.38%.

The reactor can be optimized using the same NN as that used for inverse modeling, adding new variables that account for the better performance and efficiency of the reactor, such as production cost, initiator consumption, heat generation rate and others. The introduction of these variables helps to transform a Class III NN into a Class II NN, reducing the complexity of the system and helping to simplify training.

Adding optimization variables, such as production rate or production cost, to the NN, helps to avoid systems with multiple answers. In the case shown in this section, due to the nature of the polymerization system it is possible to have two or more operating conditions giving the same polymer characteristics (molecular weight, polydispersity and copolymer composition). If solely these characteristics are given to the NN when using it, the NN will most probably return a valid operating condition that will produce the desired polymer. Other conditions may exist, but

will not be outputted by the NN. Adding the variable production rate in the NN avoids the multiple answer problem since most probably the response will be unique. Two or more conditions may produce a given polymer characteristic, but their production rates will be different, and therefore there will be a unique solution.

The same kind of work has been applied to styrene polymerization in a batch reactor and to vinyl acetate polymerization in an emulsion reactor with results similar to those in the case of the fluidized bed reactor shown previously (Fernandes, et al., 2004).

Initiator Mixture Selection

Productivity of batch processes is related to the reduction in time required to complete each batch. An increase in productivity can be achieved by running the polymerization isothermally using a mixture of initiators with different decomposition rates. Industrial-scale reactors are designed to withstand a maximum rate of heat release by exothermic polymerization, which normally corresponds to the auto-acceleration of the polymerization rate. Nevertheless, the average rate of heat release during

the batch time is significantly lower than the maximum cooling capacity of the system, meaning that the cooling system is underutilized during most of the polymerization. The amount of heat that could still be released is represented by the gray region in Figure 11.

This potential heat can come from an increase in the polymerization rate at the beginning of the batch, which can be achieved using an initiator with a short decomposition time. Two other initiators with medium and long decomposition times can be used to spread the polymerization rate and heat release over the batch time. Thus, the amount of each initiator in the mixture can be optimized in order to increase productivity, while not exceeding the maximum heat release for which the cooling system is capable of compensating.

In a study of styrene polymerization, neural networks were used to discover the optimum operating condition of the reactor, aiming for the best mixture of initiators that could be used. The neural network shown in Figure 12 was used to evaluate whether NNs could be applied to this kind of optimization problem.

Table 4: Example of neural network prediction for fluidized bed reactor conditions.

Desired Polymer	Simulation Operating Conditions
Mw = 110900 g/mol Polydispersity = 3.4 Composition = 24% 1-Butene Production = 7.5 ton/h	Ethylene Conc. = 0.3436 mol/L 1-Butene Conc. = 0.1475 mol/L Superficial Velocity = 0.29 m/s Catalyst Feed Rate = 0.075 g/s Porosity = 0.4
Neural Network Prediction Operating Conditions	**NN Prediction Error**
Ethylene Conc. = 0.3429 mol/L 1-Butene Conc. = 0.1479 mol/L Superficial Velocity = 0.289 m/s Catalyst Feed Rate = 0.0736 g/s Porosity = 0.404	Ethylene Conc. = 0.20 % 1-Butene Conc. = 0.27 % Superficial Velocity = 0.34 % Catalyst Feed Rate = 1.87 % Porosity = 1.00 %

Table 5: Example of neural network prediction for fluidized-bed reactor conditions.

| Desired Polymer	Simulation Operating Conditions
Mw = 99000 g/mol Polydispersity = 3.3 Composition = 24% 1-Butene Production = 5.1 ton/h	Ethylene Conc. = 0.2907 mol/L 1-Butene Conc. = 0.1246 mol/L Superficial Velocity = 0.232 m/s Catalyst Feed Rate = 0.100 g/s Porosity = 0.7
Neural Network Prediction Operating Conditions	**NN Prediction Error**
Ethylene Conc. = 0.2908 mol/L 1-Butene Conc. = 0.1250 mol/L Superficial Velocity = 0.233 m/s Catalyst Feed Rate = 0.098 g/s Porosity = 0.699	Ethylene Conc. = 0.60 % 1-Butene Conc. = 0.03 % Superficial Velocity = 0.38 % Catalyst Feed Rate = 1.75 % Porosity = 0.11 %
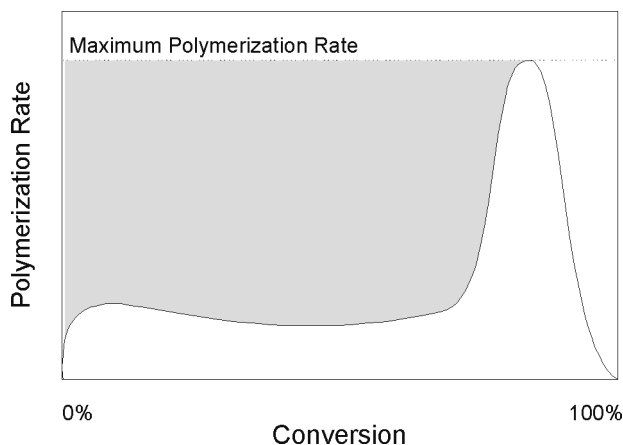
Figure 11: Polymerization rate.
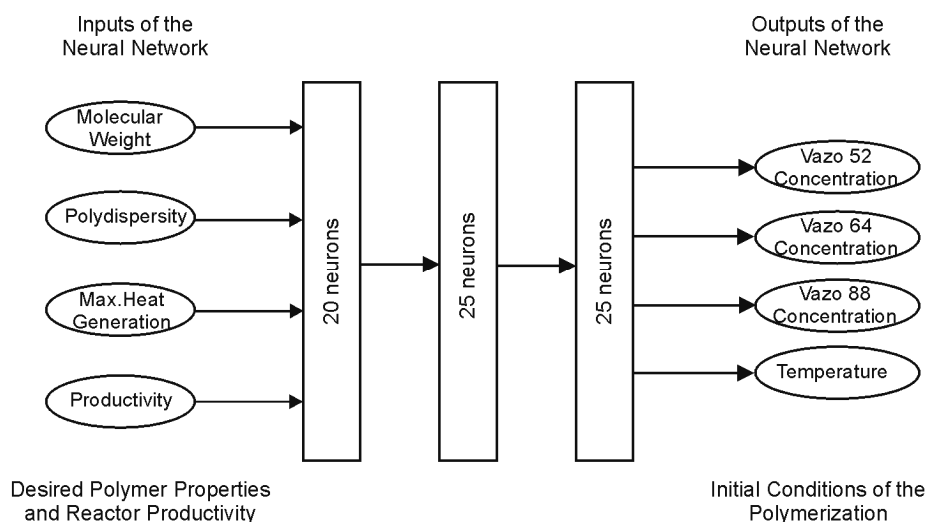
Figure 12: Class **II** NN used in selection of the initiator mixture for styrene bulk
polymerization (number of neurons shown refers to the best topology encountered).

Three initiators were selected and used in the initiator mixture: Vazo 52, Vazo 64 and Vazo 88. The decomposition rate constants for these initiators are shown in Table 6.

The data used to train the NNs were obtained with a mathematical model for bulk polymerization of styrene. Table 7 presents the ranges of temperature and initiator concentration that were used in the simulations. The operating conditions (initiator concentration and temperature) were selected randomly from the range presented in Table 7. Randomly selected data rather than factorially designed data were used since random data provides better training for this kind of NNs. A total of 394 operating conditions were simulated, with 298 used to train the NN and 96 to test it. Prediction errors for each variable after NN training are presented in Table 8.

Formulation of an optimal initiator mixture can be stated as an optimization problem in which the decision variables are the amount of each initiator and the operating temperature. The constraints to be satisfied include the final desired quality of the polymer (molecular weight and polydispersity), maximum cooling capacity and desired productivity. The results that were obtained were promising, and the prediction errors using NN were small. Table 9 shows typical examples of the prediction that can be made for this problem.

Besides their use in the selection of initiator mixtures, trained NNs can be used to optimize reactor productivity and polymer quality as well. Productivity can be improved using the NN to search for new operating conditions that, for example, can increase productivity while maintaining all other polymer

characteristics constant. Figure 13 shows an example of the increase in productivity that can be obtained using the NN.

In order to optimize productivity, a known case (operating condition 1) was used as the starting point for the optimization. A search procedure was created to find the optimum point, which consisted of increasing the productivity variable (NN input variable) while maintaining constant the values of the other input variables. Upon each increase, the trained NN outputted new operating conditions for the reactor in order to achieve that specified productivity. The increase in the value of productivity continued till an invalid value for the operating conditions was outputted by the NN, marking the end of the search for optimum productivity. The invalid value can be an impossible operating condition (such as a negative concentration) or a condition outside the training range.

Table 6: Decomposition constants for Vazo initiators.

Initiator	Pre-exponential Factor [min^{-1}]	Activation Energy [J/mol]
Vazo 52	$6.25 \cdot 10^{10}$	88578.0
Vazo 64	$4.22 \cdot 10^{10}$	84319.0
Vazo 88	$3.74 \cdot 10^{10}$	93561.0

Table 7: Ranges used for each operating condition.

Variable	Range	
	Minimum	Maximum
Initiator Concentration [mol/L]	0.008	0.1
Temperature [K]	333	383

Table 8: Prediction errors for the final NN.

Topology	Overall	Mean Prediction Errors [%]			
	Error	Vazo 52	Vazo 64	Vazo 88	Temperature
4-20-25-25-4	0.091	3.80	2.62	2.05	0.17

Table 9: Selection of initiator mixtures for styrene bulk polymerization.

Desired Polymer and Constraints	Predicted Operating Conditions	Prediction Error (%)
Mw = 106150 g/mol	Vazo 52 = 0.093 mol/L	1.70
Polydispersity = 18.4	Vazo 64 = 0.095 mol/L	0.02
Maximum Heat = 53.2 kJ/L.min	Vazo 88 = 0.070 mol/L	2.37
Productivity = 0.007 kg/L.min	Temperature = 334.7 K	0.29
Mw = 75850 g/mol	Vazo 52 = 0.063	1.11
Polydispersity = 11.2	Vazo 64 = 0.070 mol/L	1.60
Maximum Heat = 39.1 kJ/L.min	Vazo 88 = 0.082 mol/L	1.43
Productivity = 0.012 kg/L.min	Temperature = 356.1 K	0.64
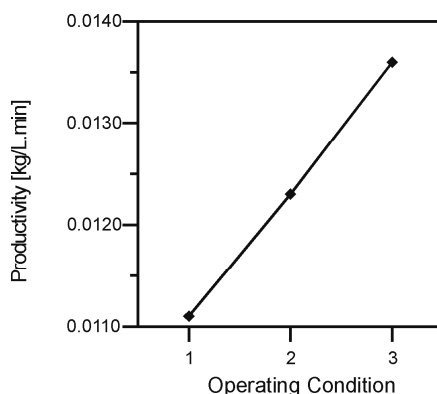
Operating Conditions	1	2	3
Productivity [kg/L.min]	0.0111	0.0123	0.0136
Temperature [K]	355	354	358
Vazo 52 [mol/L]	0.039	0.062	0.047
Vazo 64 [mol/L]	0.079	0.069	0.067
Vazo 88 [mol/L]	0.030	0.083	0.054

Figure 13: Optimization of productivity

Grade Changes

When dealing with continuous reactors, NNs can be useful in designing grade changes. Profiles for temperature and concentration can be inferred from the current grade, the target grade and the kind of switchover to be followed during the change. A study of grade changes was done for vinyl acetate emulsion polymerization in continuous reactors (Fernandes, et al., 2004).

Emulsion polymerization of vinyl acetate is a heterogeneous reaction system, in which a basic batch recipe is composed of monomer (vinyl acetate), water, initiator and emulsifier. Thus, when running a batch reaction to polymerize vinyl acetate four variables must be set as the reactor's operating conditions: monomer, initiator and emulsifier concentrations and temperature. Operation under these conditions will produce a polymer with a given molecular weight, polydispersity, particle diameter and branching frequency, so a NN for the system will have input and output variable as in the structure shown in Figure 14.

The data used to train the NN were obtained running a mathematical model for emulsion polymerization of vinyl acetate and some experimental data points were also used. The ranges of temperature and initiator concentration that were used in the simulations are presented in Table 10. The operating conditions (initiator concentration and temperature) were selected randomly from the ranges presented in Table 10. A total of 394 operating conditions were simulated, with 298 used to train the NN and 96 to test and cross-validate it.

A NN with a 5-20-25-4 topology was selected as the most suitable network and was trained for 250000 iterations, lowering the mean prediction errors to less than 5% for all variables (Table 11).

Grade changes can be studied by setting a target grade for the polymer (molecular weight, polydispersity, branching frequency, particle diameter and productivity) and using the NN to predict the operating conditions in order to produce the target grade, as shown in Table 12.
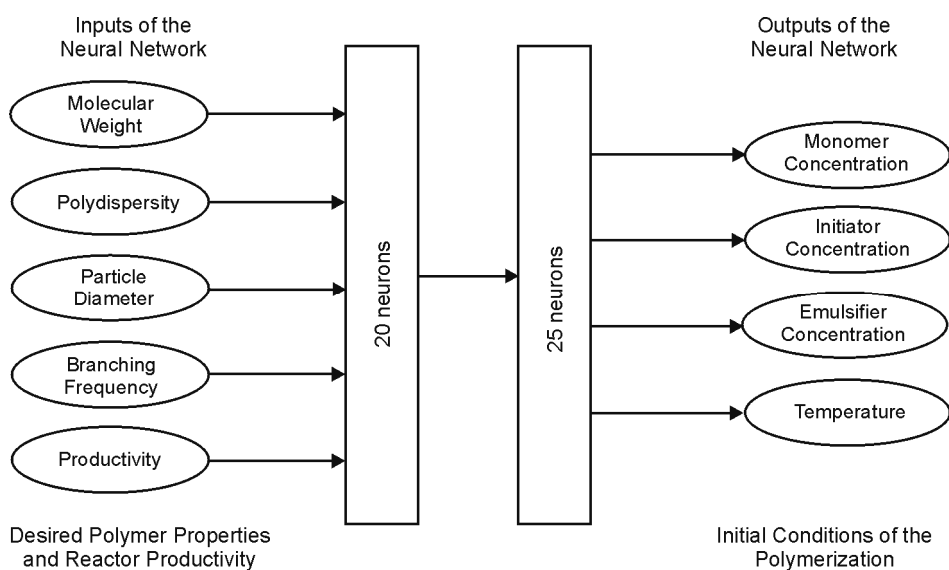

Figure 14: Class **I** NN used in the emulsion polymerization of vinyl acetate
(number of neurons shown refers to the best topology encountered).

Table 10: Ranges used for each operating condition.

	Training Range		Operating Range	
	Minimum	**Maximum**	**Minimum**	**Maximum**
Vinyl Acetate [mol/L water]	1.50	5.00	1.80	4.00
Initiator [mol/L water]	0.0007	0.070	0.001	0.050
Emulsifier [mol/L water]	0.020	0.200	0.030	0.150
Temperature [K]	318	358	323	353

Table 11: Prediction errors.

Topology	Overall	Mean Prediction Errors [%]			
		Monomer	Initiator	Emulsifier	Temperature
5-20-25-4	0.033	0.5	2.8	1.0	0.1
	Best Prediction	0.6	0.2	0.5	0.1

Table 12: Grade change for emulsion polymerization of vinyl acetate.

Current Grade	Target Grade
Mw = 3000000 g/mol	Mw = 2500000 g/mol
Polydispersity = 5.5	Polydispersity = 4.7
Branching = 0.0016	Branching = 0.0013
Particle Diameter = 100 nm	Particle Diameter = 95 nm
Productivity = 0.09 g/L.s	Productivity = 0.06 g/L.s
Operating Conditions for the Current Grade	**Operating Conditions for the Target Grade**
VAc = 5.76 mol/L water	VAc = 4.04 mol/L water
Initiator = 0.0044 mol/L water	Initiator = 0.0017 mol/L water
Emulsifier = 0.0545 mol/L water	Emulsifier = 0.0473 mol/L water
Temperature = 328.0 K	Temperature = 333.2 K

Different grade change policies can be simulated. Figure 15 shows such a grade change if a hypothetical linear change is desired for the polymer properties and production rate, i.e., polymer quality and production change at a constant rate. In this case, the changes in the operating conditions are smooth, and hence the likelihood of policy feasibility is enhanced.

If a smaller amount of off-specification product is desired, a policy that changes the production rate rapidly can be devised. Figure 16 shows the profile for the example of production rate to be employed in the grade change and the profiles for the operating conditions that should be used for the grade change. In this case, the changes in the operating conditions are also smooth, following a feasible path that can be implemented and easily controlled.

The kind of grade change can be optimized by comparing the results from the NN for total cost, amount of off-spec product produced and shorter time of grade changing, among others.
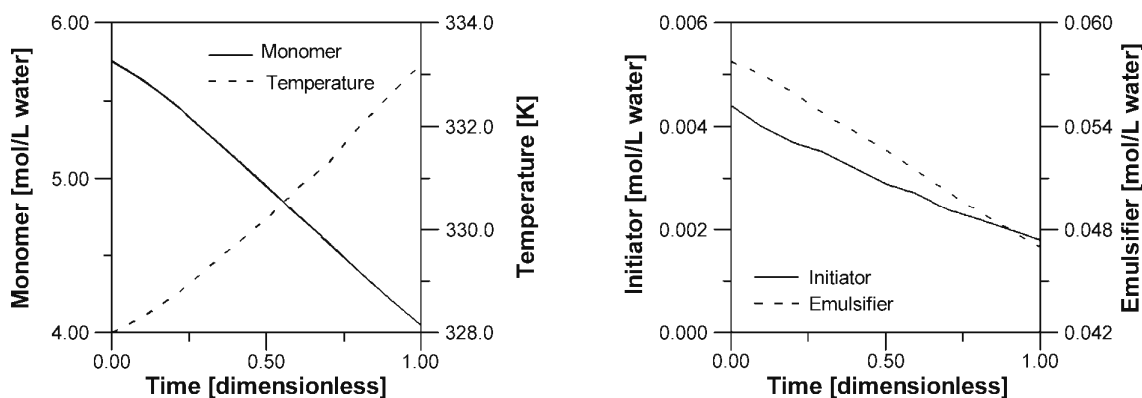
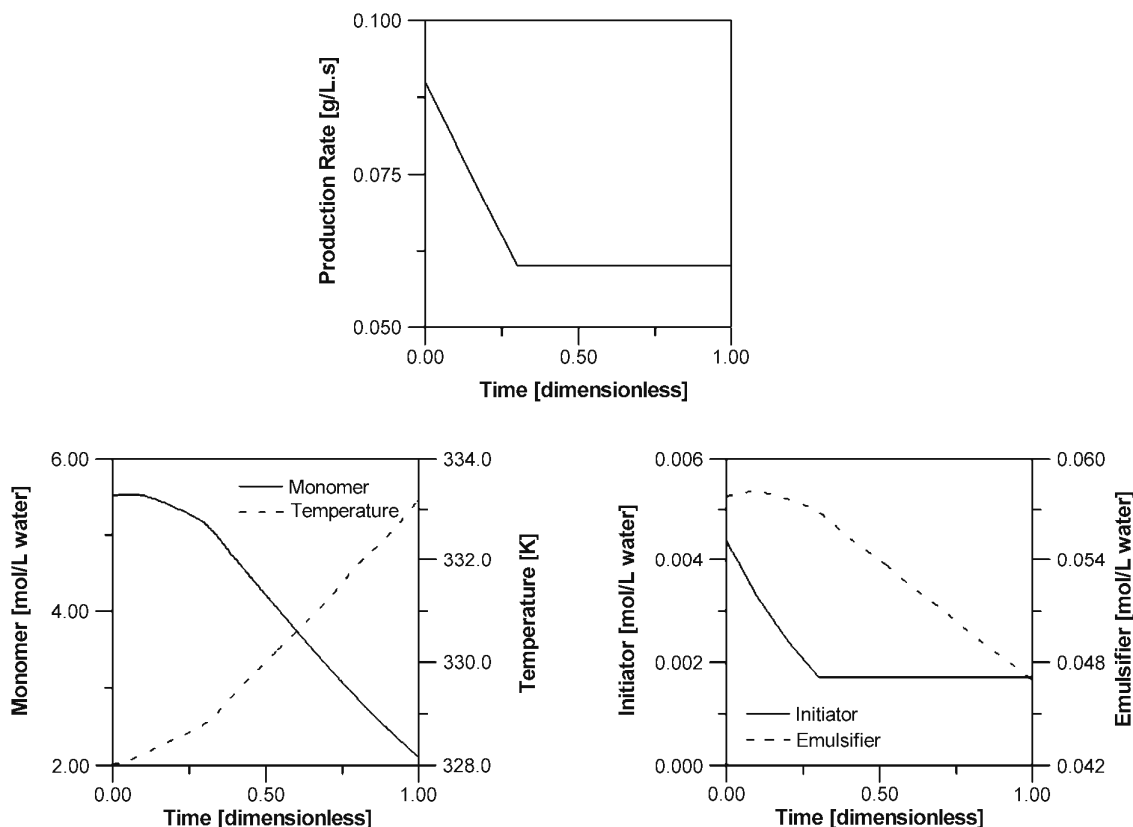**Figure 15:** Operating conditions for a smooth linear grade change.

Figure 16: Operating conditions for a grade change aiming at
smaller amount of off-specification products.

Process and Quality Control

Quality control in batch processes is challenging because product quality is not known until batch processing has been completed and because a direct measurement of the molecular weight and molecular weight distribution of the polymer is not available instantaneously. In general these measurements are made indirectly (by means of the viscosity and density of the reaction media) or directly but in this case there is a time delay between the time at which the sample is taken and the time at which the result is available. Two good ways of dealing with the problem are available in the literature. The first uses a hybrid NN (Tsen et al., 1996) and the second uses the recurrent NN (Tian et al., 2002).

When using hybrid NNs the initial condition of the reaction (at t = 0) is sent to the NN, along with the results of an intermediate measurement of polymer quality taken at t = x. The NN processes this information and returns the new operating conditions to the reactor in order to compensate for the any deviation (or disturbance) in the process. The new conditions can imply a new temperature or addition of monomers or initiators. This procedure is recommended when one or more intermediate measurements of polymer quality are taken during the batch.

When on-line measurements of viscosity, density, monomer concentration or other variables are available, then the use of a recurrent NN is recommended, since this will permit better quality control.

CHALLENGES IN THE FIELD OF POLYMERIZATION

Many challenges still await solution in the use of NN in the field of polymerization. Some challenges can be pinpointed:

- To achieve a better understanding of how the topology of the NN affects the prediction results, especially for the number of hidden layers and Classes II and III NNs.
- To developed better ways of training the NN in the shadow zone.
- To conduct studies of inverse modeling for more complex reactors and reactions, especially for copolymerizations and emulsion reactors.

- To find a NN that better deals with problems that have multiple answers, like the inverse modeling problem.
- To improve the NNs ability to handle a large number of recipes. In controlling polymerization reactors, the procedures in use today handle only one recipe per NN.
- Inverse modeling using advanced NNs, such as fuzzy NNs.

CONCLUSIONS

The great challenge in NN research is to come up with better procedures for the use of NNs. Currently, the application of NNs requires of the researcher (or user) a good knowledge of NNs and of the process for which the NN will be used. Finding the best topology is still very time consuming and can sometimes be frustrating, resulting in bad predictions.

In this paper a brief tutorial of the usual and recommended practices to be used with NNs were presented and some practical tips based on our experience were given. Several cases were also reviewed showing ways that NNs can be applied with success.

The biggest challenge now is to apply NNs to complex problems, especially those related to quality control, complex process optimization and inverse modeling. These applications require more complex NNs such as Class II and Class III NNs, which often require two or more hidden layers and a large number of neurons in each hidden layer.

Neural networks have good potential for use in the field of polymerization but Class II and Class III NNs still need to be better understood and their predictions to become more reliable and precise; this is the great challenge to the scientific community now so that the use of NNs can be advanced.

ACKNOWLEDGEMENTS

NOMENCLATURE

a	parameter of the sigmoid function
Y_{max}	maximum value of variable Y
Y_{min}	minimum value of variable Y
Y_{norm}	normalized variable Y

REFERENCES

Al-Haik, M.S., Garmestani H. and Savran A., Explicit and Implicit Viscoplastic Models for Polymeric Composite, Int.J.Plasticity, 20, 1875 (2004).

Chan, W.M. and Nascimento, C.A.O., Use of Neural Networks for Modeling of Olefin Polymerization in High Pressure Tubular Reactors, J.Appl.Pol.Sci., 53, 1277 (1994).

Curry, B., Morgan, P. and Beynon, M., Neural Networks and Flexible Approximations, IMA Journal of Mathematics Applied in Business and Industry, 11, 19 (2000).

Curry, B., Morgan, P. and Silver, M., Neural Networks and Non-Linear Statistical Methods: an Application to the Modelling of Price-Quality Relationships, Comp.Op.Res., 29, 951 (2002).

Ebube, N.K., Ababio, G.O. and Adeyeye, C.M., Preformulation Studies and Characterization of the Physicochemical Properties of Amorphous Polymers using Artificial Neural Networks, Int.J.Pharm., 196, 27 (2000).

Fernandes, F.A.N., Modelagem e Simulação de Reatores de Polimerização e Caracterização de Polímeros. Ph.D. diss., Unicamp-Campinas, Brazil (2002).

Fernandes, F.A.N.; Lona, L.M.F., Application of Neural Networks on the Definition of the Operational Conditions of Fluidized Bed Polymerization Reactors, Pol.Reac.Eng., 10, 181 (2002).

Fernandes, F.A.N., Lona, L.M.F. and Penlidis, A., Inverse Modeling Applications in Emulsion Polymerization of Vinyl Acetate, Chem.Eng.Sci., 59, 3159 (2004).

Fraser, C.M., *Neural Networks: Literature Review from a Statistical Perspective*. Statistics Department, California State University, Hayward (2000).

Fujii, M., Kushida, K., Ihori H. and Arii K., Learning Effect of Composite Conducting Polymer, Thin Solid Films, 438-439, 356, (2003).

Hanai, T., Ohki, T., Honda, H. and Kobayashi, T., Analysis of Initial Conditions of Polymerization Reaction using Fuzzy Neural Networks and Genetic Algorithm, Comp.Chem.Eng., 27, 1011 (2003).

Haykin, S., *Neural Networks: A Comprehensive Foundation*, Prentice Hall, New York (1998).

Hinchliffe, M., Montague, G., Willis, M. and Burke A., Correlating Polymer Resin and End-Use Properties to Molecular-Weight Distribution, AIChE Journal, 49, 2609 (2003).

Hornik, K., Stinchcombe, M. and White, H., Multilayer Feedforward Networks are Universal Approximators, Neural Networks, 2, 359 (1989).

Huang, H.X. and Liao, C.M., Prediction of Parison Swell in Plastics Extrusion Blow Molding using a Neural Network Method., Polym.Testing, 21, 745 (2002).

Krothapally, M. and Palanki, S., A Neural Network Strategy for Batch Process Optimization, Comp.Chem.Eng., 21, S463 (1997).

Kuroda, C. and Kim, J., Neural Network Modeling of Temperature Behavior in an Exothermic Polymerization Process, Neurocomputing, 43, 77 (2002).

Morgan, P., Curry, B. and Beynon, M., Neural Networks Approximations for Different Functional Forms, Expert Systems, 16, 2 (1999).

Nascimento, C.A.O., Giudici, R. and Guardani, R., Neural Network Based Approach for Optimization of Industrial Chemical Processes, Comp.Chem.Eng., 24, 2303 (2000).

Simon, L. and Fernandes, M., Neural Network-Based Prediction and Optimization of Estradiol Release from Ethylene–Vinyl Acetate Membranes, Comp.Chem.Eng., 28, 2407 (2004).

Sumpter, B.G. and Noid, D.W., Neural Networks and Graph Theory as Computational Tools for Predicting Polymer Properties, Macromol. Theor. Sim., 3, 363 (1994).

Sun, Q., Zhang, D., Chen, B. and Wadsworth, L.C., Application of Neural Networks to Meltblown Process Control., J.Appl.Pol.Sci., 62, 1605 (1996).

Syu, M.J. and Tsao, G.T., Neural Network Modeling of Batch Cell Growth Pattern. Biotech. & Bioeng., 42, 376 (1993).

Tamura, S. and Tateishi, M., Capabilities of a Four-Layered Feedforward Neural Network: Four Layers Versus Three, IEEE Trans.Neural Net., 8, 251 (1997).

Tian, Y., Zhang, J. and Morris, J., Optimal Control of a Batch Emulsion Copolymerization Reactor Based on Recurrent Neural Network Models, Chem.Eng.Proc., 41, 531 (2001).

Tsen, A.Y.D., Jang, S.S., Wong, D.S.H. and Joseph, B., Predictive Control of Quality in Batch Polymerization Using Hybrid ANN Models, AIChE Journal, 42, 455 (1996).

White, H., *Artificial Neural Networks: Approximation and Learning Theory*, Blackwell, New York (1992).

Wold, S., Cross-Validatory Estimation of the Number of Components in Factor and Principal Components Models, Technometrics, 20, 397 (1978).

Xiong, Z. and Zhang, J., A Batch-to-Batch Iterative Optimal Control Strategy Based on Recurrent Neural Network Models, J.Proc.Control, 15, 11 (2005).

Zhang, J., Inferential Estimation of Polymer Quality using Bootstrap Aggregated Neural Networks, Neural Net., 12, 927 (1999).

Zhang, Z., Barkoula, N.M., Karger-Kocsis J. and Friedrich K., Artificial Neural Network Predictions on Erosive Wear of Polymers, Wear, 255, 708 (2003).

Zhang, J., Martin, E.B., Morris A.J. and Kiparissides C., Inferential Estimation of Polymer Quality using Stacked Neural Networks, Comp. Chem. Eng., 21, S1025 (1997).