7 APPLICATION OF NEURAL NETWORKS IN OIL REFINERIES

Ali Zilouchian and Khalid Bawazir

7.1 INTRODUCTION

In response to demands for increasing oil production levels and more stringent product quality specifications, the intensity and complexity of process operations at oil refineries have been exponentially increasing during the last three decades. To alleviate the operating requirements associated with these rising demands, plant designers and engineers are increasingly relying upon automatic control systems. It is well known that model based control systems are relatively effective for making local process changes within a specific range of operation [1]. However, the existence of highly nonlinear relationships between the process variables (inputs) and the product stream properties (outputs) have bogged down all efforts to come up with reliable mathematical models for large scale crude fractionation sections of an oil refinery. In addition, the old inferred property predictors are neither sufficiently accurate nor reliable for utilization of advanced control applications [2]. On the other hand, the implementation of intelligent control technology based on soft computing methodologies such as neural network (NN), fuzzy logic (FL), and genetic algorithms (GA) can remarkably enhance the regulatory and advanced control capabilities of various industrial processes such as oil refineries [3]-[11].

Presently, in the majority of oil refineries (such as Ras Tanura located in Saudi Arabia), product samples are collected once or twice a day according to the type of analysis to be performed and supplied to the laboratory for analysis. If the laboratory results do not satisfy the specifications within an acceptable tolerance, the product has to be reprocessed to meet the required specification [2]. This procedure is costly in terms of time and dollars. In the first phase, an off-line specification product should be first routed to a holding facility. In the second phase, the process should be tuned before any further processing is carried out. In order to resolve this problem in a timely fashion, a continuous on-line method for predicting product stream properties and consistency with and pertinence to column operation of the oil refinery are needed.

In general, on-line analyzers can be strategically placed along the process vessels to supply the required product quality information to multivariable controllers for fine tuning of the process. However, on-line analyzers are very costly and maintenance intensive. To minimize the cost and free maintenance resources, alternative methods should be considered.

In this chapter, the utilization of artificial neural network (ANN) technology for the inferential analysis of a crude fractionation section of the Ras Tanura Oil Refinery at Dhahran is presented. The implementation of several neural network models using back propagation algorithm based on collection of real-time data for a three-months operation, of the plant is presented. The proposed neural network architectures can accurately predict various properties associated with crude oil production. The simulation results for modeling of several products such as naphtha 95% cut point and naphtha Reid vapor pressure are analyzed. The results of the proposed work can ultimately enhance the on-line prediction of crude oil product quality parameters for the crude fractionation processes of various oil refineries.

The chapter is organized as follows. Section 7.2 covers various steps pertaining to collection of plant data that are used during the training and verification phases of the neural network program. A systematic procedure to construct a NN model is also presented in this section. In section 7.3, selection of appropriate data sets as well as data analysis procedures are discussed. Section 7.4 is devoted to various steps in the implementation phase of neural network models in the crude oil fractionation process. In sections 7.5 and 7.6, the training procedures as well as the results of modeling for naphtha 95% Cut-point and naphtha Reid vapor pressure products are analyzed. It is shown that the proposed NN models predict products qualities well within the specified error goals in both training and verification phases. Various implementation issues such as model building, model data analysis, effects of neuron distribution on training, and model robustness are also discussed in this section. Finally, section 7.6 summarizes the contributions of this chapter.

7.2 BUILDING THE ARTIFICIAL NEURAL NETWORK

The mathematical algorithms developed to model neurons can be adapted for many useful predictions in processing plants. The complexity of the pattern to be recognized dictates the complexity of the required algorithm. Some very useful predictions can be constructed in processing plants using algorithms whose coefficients are discovered through training. Figure 7.1 is a graphical representation of the artificial neural network structure.

A neural network predictor is built by discovering the weights as shown in Figure 7.1. N_1K_1 through N_1K_n are the corresponding weights of the first neuron. The output Q_p is the predicted inferred process stream property (%H_2S, 95% cut-point, etc.)The coefficients of the model are discovered by training a neural network program using back propagation algorithms [12, 13, 17]. The inputs of NN consist of plant data such as temperature, flow rate and pressure where, the respective product quality is considered as desired output of the program model. The neural network program will be trained by adjusting the weight coefficients until the difference between the predicted product quality and the measured product quality is within acceptable limits. When the coefficients have been determined, they should be tested by comparing the predicted quality

to the measured quality for data sets which were not used in finding the coefficients. The process of finding the ANN coefficients is called training the network [13], [14].

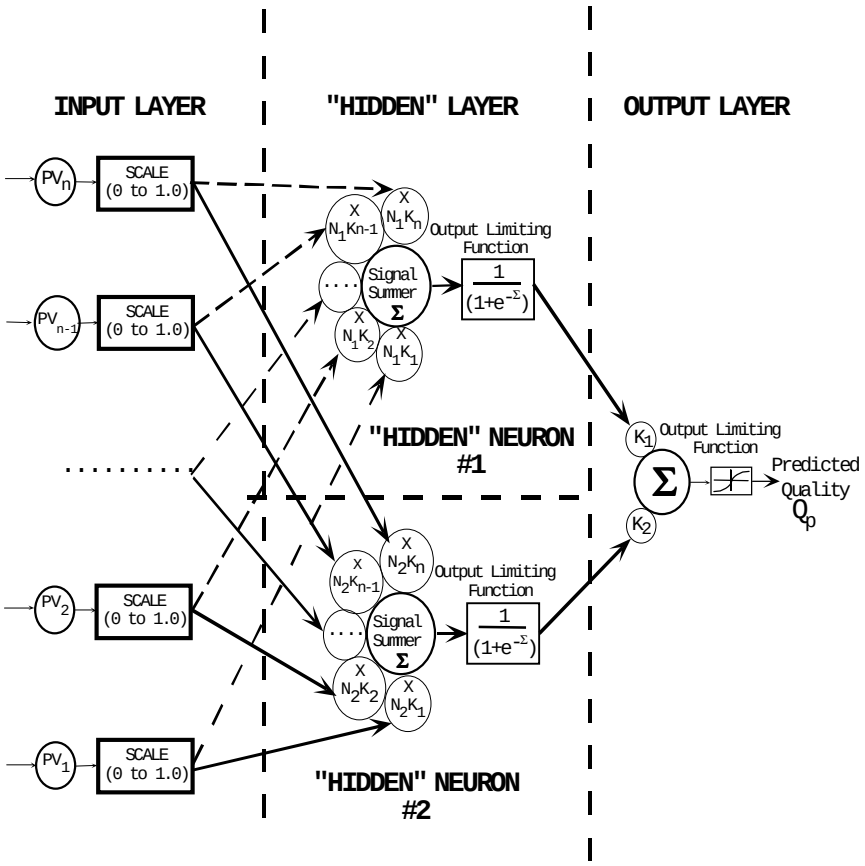

Figure 7.1: Graphical representation of ANN structure.

7.2.1 Range of Input Data

Neural networks will not be an accurate predictor if the operating inputs/output data are outside their training data range. Therefore, the training data set should possess sufficient operational range including the maximum and minimum values for both input/output variables.

7.2.2 Size of the Training Data Set

A minimum of two valid data sets is required for each coefficient in the training algorithm. A large number of valid data sets provide much better accuracy in the prediction phase. However, some training data sets are not valid either due to the dynamic nature of a process or as the result of inaccuracies in data acquisition techniques. A large data set will average out various inaccuracies within a system.

7.2.3 Acquiring the Training Data Set

The least intrusive technique for obtaining the training data set is to take data during the course of normal operations. This procedure probably will not satisfy the required variations in some process variables. However, plant tests can be accomplished by varying the process variables within the region of interest to complete the gaps within the required data. In general, it is not necessary to have field analyzers to develop a neural network predictor for a stream quality. Samples can be taken and sent to the laboratory for analysis at the same time that data (flow, temperature, and pressure) are taken from field transmitters.

7.2.4 Validity of the Training Data Set

In many industrial applications accuracy is not as important as repeatability. For example, a network trained for a pressure transmitter with a 15 pound per square inch (PSI), zero shift will predict accurately - unless the transmitter is re-calibrated; however, the lack of repeatability exhibited in data taken from hand-written shift logs has proven too unrepeatable to use as a training set. The training data set can be taken from the Distributed Control System (DCS) or the supervisory control and data acquisition computer.

Many of the processes have significant time constants and dead times [8]. Unless it is desired to include these time constants and dead times in the prediction, the process should have been operating at steady-state for a period equal to at least two time constants before including the operating data in the data set. Flow and pressure inputs should be averaged to eliminate the problem of signal noise.

7.2.5 Selecting Process Variables

Initial process variable selection is not critical; almost anything upstream of the measurement point could be useful. As many process variables should be included as can be handled. The training process will automatically determine which are important and which can be deleted from the calculation.

For example, the process variables shown in Figure 7.2 are selected to predict Reid vapor pressure (RVP) in the bottom of a stripper column. Their relative importance, determined by neural network training, is shown in Table 1. If those process variables chosen initially do not give the required accuracy of

prediction, less important variables should be dropped and other parameters added.

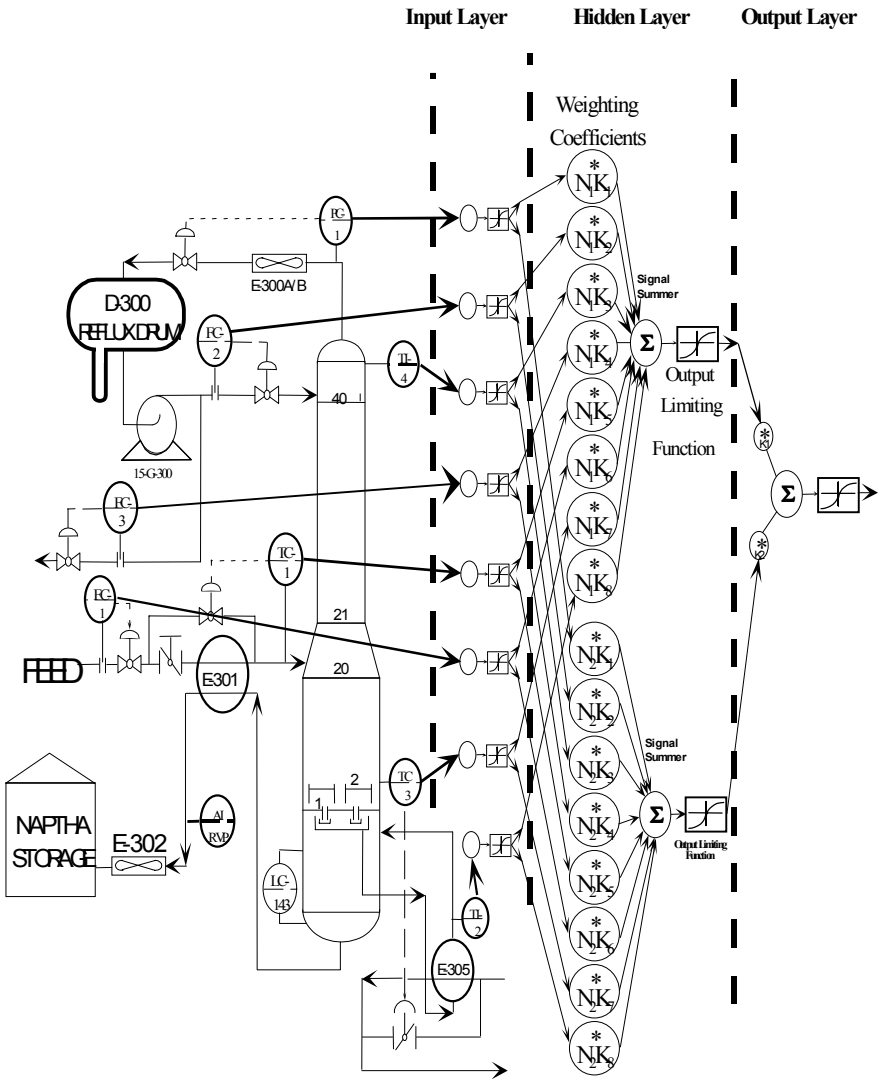

Figure 7.2: Process Variable Selection in the Stripper Column.

Table 7.1: Relative Importance of Process Variables Determined by NN

FEED FLOW RATE	F-1	4 %
REFLUX FLOW RATE	F-2	8 %
OVERHEAD PRESSURE	P-1	6 %
FEED TEMPERATURE	T-1	13 %
REBOILER RETURN TEMPERATURE	T-2	27 %
TRAY 2 TEMPERATURE	T-3	36 %
COLUMN TOP TEMPERATURE	T-4	5 %
DISTILLATE FLOW RATE	F-3	2 %

7.3 DATA ANALYSIS

The first step in data analysis is to ensure that all column parameters are collected properly. Data unavailable due to transmitter downtime or calibration at the time of data collection should be identified. Since artificial neural networks require that all model parameters be available all the time, unavailable data for any of the parameters force the elimination of the complete data set that is collected at that time. This includes lab data, which is not collected at the scheduled sampling time. The definition of a complete data set includes all process parameters plus one lab value. Lab analyzed properties (95% cut-point, sulfur content, freeze point, etc.) are analyzed individually to generate neural network models.

Once a complete set of parameters is collected, the neural network model can then be used to do a complete data analysis. the neural network model allows the user to specify which data set will be used for model building (teaching phase), and which for model verification (testing phase). A statistical method can be used to eliminate a suspected bad lab data set. The main assumption of the statistical method is that there has to be a correlation between model inputs (process parameters) and model output (lab value).

7.3.1 Elimination of Bad Lab Values

Bad lab values can be identified as follows: The neural network model is given three data sets for model verification (out of 180 data sets), and the rest of the data sets are used for model building. All data sets are switched between model verification mode to model building mode until all data are tested. At any point during the above process, if any of the three model verification points fall outside the lab repeatability, the degree of deviation from repeatability is recorded. At the end of this analysis, all deviant points are completely removed from model building. Thus, it can be established that all remaining data sets conform to the general trend of the correlation.

As a final step, each of the deviant points is again individually added to the model and tested as a verification point by itself. If this point still falls outside

lab repeatability, then it is permanently eliminated. Otherwise, the point is returned to the model.

The elimination of data sets during this step does not necessarily reflect only bad lab value. It is possible that the lab analysis is done correctly; however, either the snapshot of the process values taken do not coincide with the time of sampling by operators, or the plant is not operating at steady state conditions at the time of sampling.

7.3.2 Process Parameters and Their Effect on NN Prediction

All identified process parameters do not necessarily have an effect on each of the lab values (properties). The final step of data analysis is to identify the most important process parameters that have a significant effect on the inferred analysis and eliminate those parameters which have little or no effect. Two methods can be used to perform the elimination process. The first is using engineering judgment to realize which process parameters can have little or no effect on the model. An example of this is to remove all naphtha stabilizer parameters when the network is being used to model riesel sulfur.

The second method is to utilize the neural network model itself. The neural network program can generate an analysis of the final weights given to each of the process parameters to fit the data. This method of elimination, however, is not as straightforward as one might expect. The neural network model relies more on process parameters with a large degree of variance. It is possible that the most important parameter that affects a particular lab data set keeps the same value in all the generated data sets. The neural network program will ignore such a parameter. Thus elimination should not include variables which, from an engineering point of view, should have a contribution on the inferred analysis.

For example, a fuel gas density analyzer in the plant under investigation gives density measurements (used for heating value) about five minutes after they would be useful to improve furnace (and the affected crude column) stability. Data for the fuel gas supply pressure, burner gas pressure (P-236), burner gas pressure (P-120), heater coil outlet temperature (T-178), and gas pressure controller output to valve (PC-120 VO) are used as the inputs for a neural network training set – with the output of the density analyzer (A-156) as the stream quality to be predicted. The block diagram of the plant is shown in Figure 7.3. The data sets are used to train a neural network predictor with five hidden sigmoidal neurons. The training result can be found in reference [3]. However, the output of NN model indicates that the algorithm does not predict the stream quality (gas density) with enough accuracy to be useful. It was concluded that an input which would prevent the predicted quality from varying when the measured quality is constant is not in the training set.

If the missing input (fuel gas flow rate) is added to the data set, the algorithm predicts gas density by using the difference in deferential pressure resulting from flow through a fixed and a variable orifice – and becomes useful for eliminating upsets introduced by rapid fuel gas density (and heat of combustion)

fluctuations. The simulation results indeed have shown an accurate model prediction after adding the fuel gas flow rate to the data set [3]. Note that it is necessary to align data from different inputs to get a data set whose elements occur simultaneously.

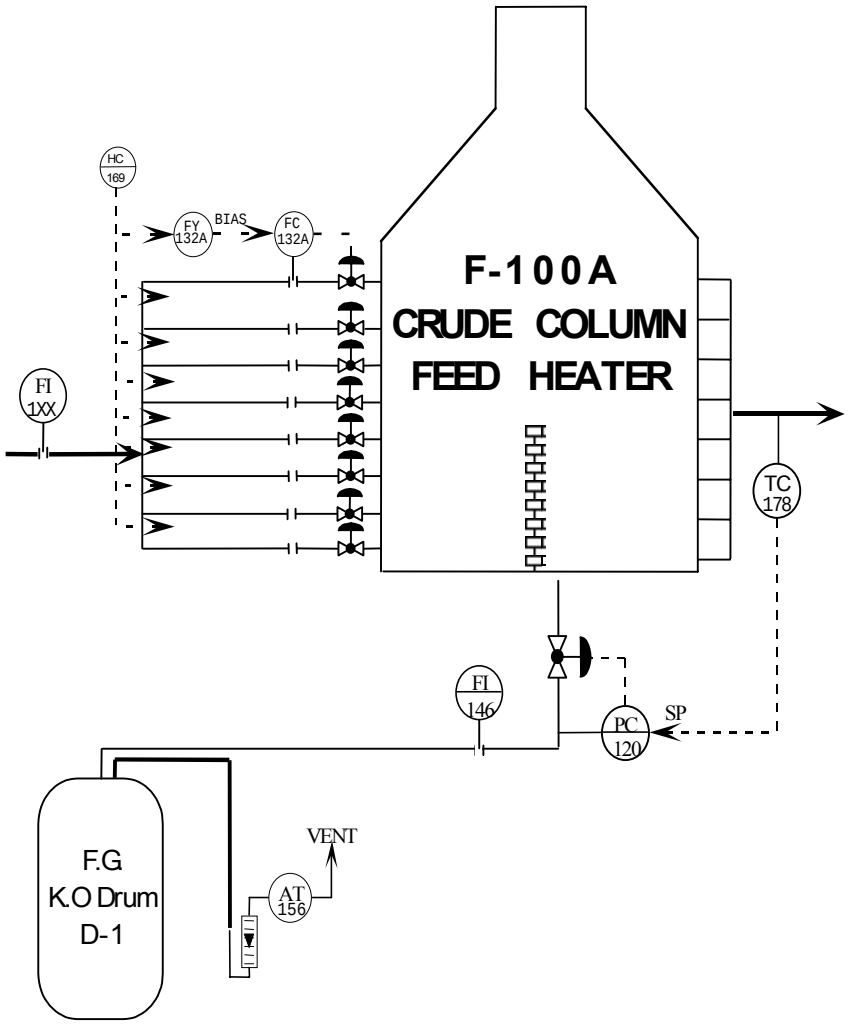

Figure 7.3: Block Diagram of Fuel Gas Supply System.
(F.G. K.O.= Fuel Gas Knock Out)

7.4 IMPLEMENTATION PROCEDURE

The major steps that are involved in implementing the ANN predictor are shown in Figure 7.4.

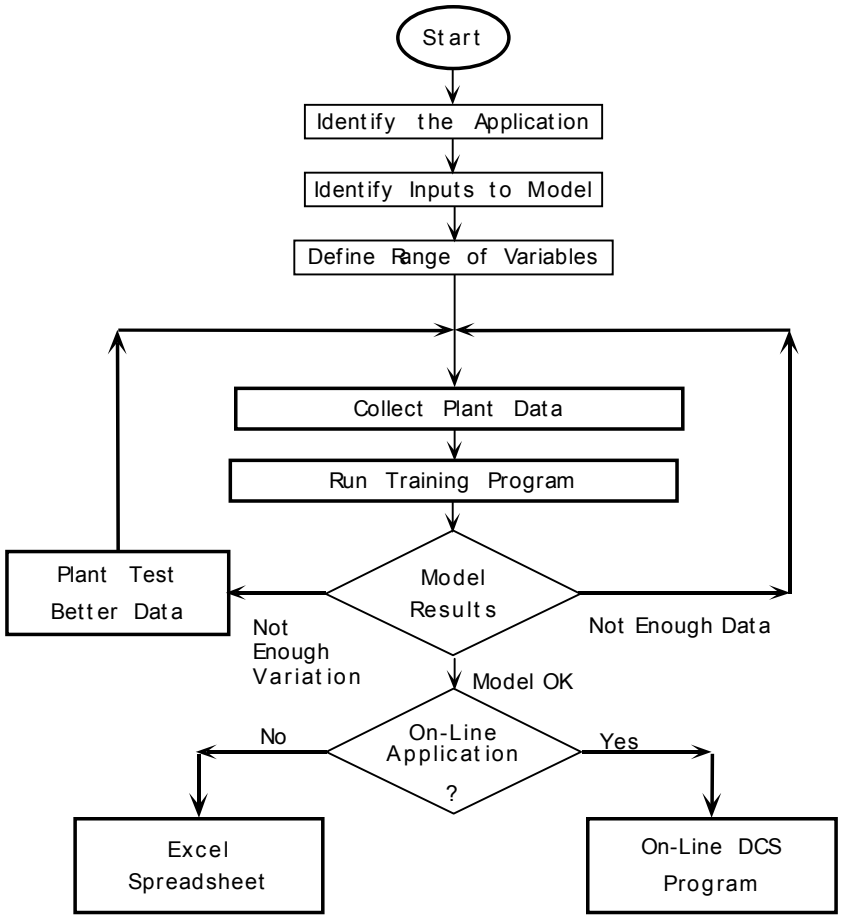

Figure 7.4: Major Steps for Implementing of ANN in the Crude Oil Fractionation Process.

7.4.1 Identifying the Application

The first step for construction of an NN model is the appropriate identification of a potential application. For example, suppose Ethane is burned as fuel gas and propane with 2% ethane can be sold for $18/barrel in the market. The operating objectives of a de-ethanizer unit will be to minimize the propane in the overhead product and maintain slightly less than 2% Ethane in the column bottoms.

On the other hand, if ethane, with 1.5% propane sells for $16/barrel and propane is used as fuel gas, the operating objectives of the de-ethanizer will be different from the above case: to minimize ethane in the column bottoms and

keep the propane in the overhead as near 1.5% as possible, without exceeding the sales limit. To achieve either set of the above objectives, a continuous measurement of propane in the column overhead and ethane in the bottoms is necessary.

7.4.2 Model Inputs Identification

The neural network algorithm will not match a random number set. For prediction model to work, there must be some relationships between input/output variables [9], [11]. Training will quantify such a relationship. If a neural network will not train with a good data set, a significant variable may not have been included in the data set. If a rigorous mathematical equation can be written between the inputs and the output, a neural network is unnecessary.

As an example, a double product (overhead and bottoms) distillation column as shown in Figure 7.5 has two variables: the heat balance and the material balance, which determine its separation capability, and two product quality variables: heavy key in the overhead (propane in the de-ethanizer overhead product) and the light key in the bottoms (ethane in the de-ethanizer bottoms). Those process measurements, flows, pressures, and temperatures, which could be used to calculate the heat and material balances, should be chosen as inputs to the neural network. To predict propane in the de-ethanizer overhead we start with the input variables as shown in Table 7.2.

Table 7.2: Process Variables for Propane Prediction
(MBD = Million barrels per day; MPPH = Thousand pounds per hour)

Process Variable	Tag	Range
Tray 27 temperature	TI-6	100-300 °F
Overhead Temperature	TI-5	100-300 °F
Reflux Temperature	TI-7	100-300 °F
Feed Temperature	TI-4	0-200 °F
Reflux Rate	FI-2	0-20MBD
Distillate Rate	FI-3	0-10MBD
Feed Rate	FI-6	0-80MBD
Reboiler Steam	FI-4	0-30MPPH
Bottoms Product	FI-5	0-80MBD

A good starting point for ethane prediction in the bottoms would use the process variables as shown in Table 7.3.

Table 7.3: Process Variables for Ethane Prediction
(MBD= Million barrels per day, MPPH= Thousands pound per hour)

Process Variable	Tag	Range
Tray 1 temperature	TI-1	100-400 °F
Bottoms Temperature	TI-3	100-400 °F
Tray 18 Temperature	TI-8	100-300 °F
Feed Temperature	TI-4	0-200 °F
Re-boiler Return Temperature	TI-2	100-400 °F
Distillate Rate	FI-3	0-10MBD
Feed Rate	FI-6	0-80MBD
Re-boiler Steam	FI-4	0-30MPPH
Bottoms Product	FI-5	0-80MBD

If the plant data include significant variation in each of these process variables and the neural network coefficients for a process variable are very small, that process variable can be dropped from the model. If the network will not train, and other conditions are met, other process variables based on engineering experience should be included in the model.

7.4.3 Range of Process Variables

The range of the process variables in the training data set should include the entire operating range. The data set should include data for each process variable, evenly distributed throughout the range for which prediction is desired.

7.5 PREDICTOR MODEL TRAINING

For the naphtha 95% cut point and naphtha Reid vapor pressure stream properties, the plant data, including the stream quality desired to predict, are collected in a Microsoft Excel™ spreadsheet to facilitate data manipulation. The data are then scaled to a fraction of the transmitter range so that they are confined to a sub-interval of [0…1]. A practical region for the data is chosen to be [0.1…0.9]. In this case each input or output parameter p is normalized as p_n before being applied to the neural network according to:

$$p_n = [(0.9 - 0.1) / (p_{max} - p_{min})] * (p - p_{min}) + 0.1$$

where p_{max} and p_{min} are the maximum and minimum values, respectively, of data parameter p. The spreadsheet file is then converted to text file and loaded into the MATLAB™ neural network toolbox [14]. The MATLAB software program uses a back propagation training algorithm to adjust the weights of the network in order to minimize the sum-squared error of the network. This is done by continually changing the values of the network weights in the direction of steepest descent with respect to the error [11]-[17]. The change in each weight is proportional to that element's effect on the sum-squared error of the network.

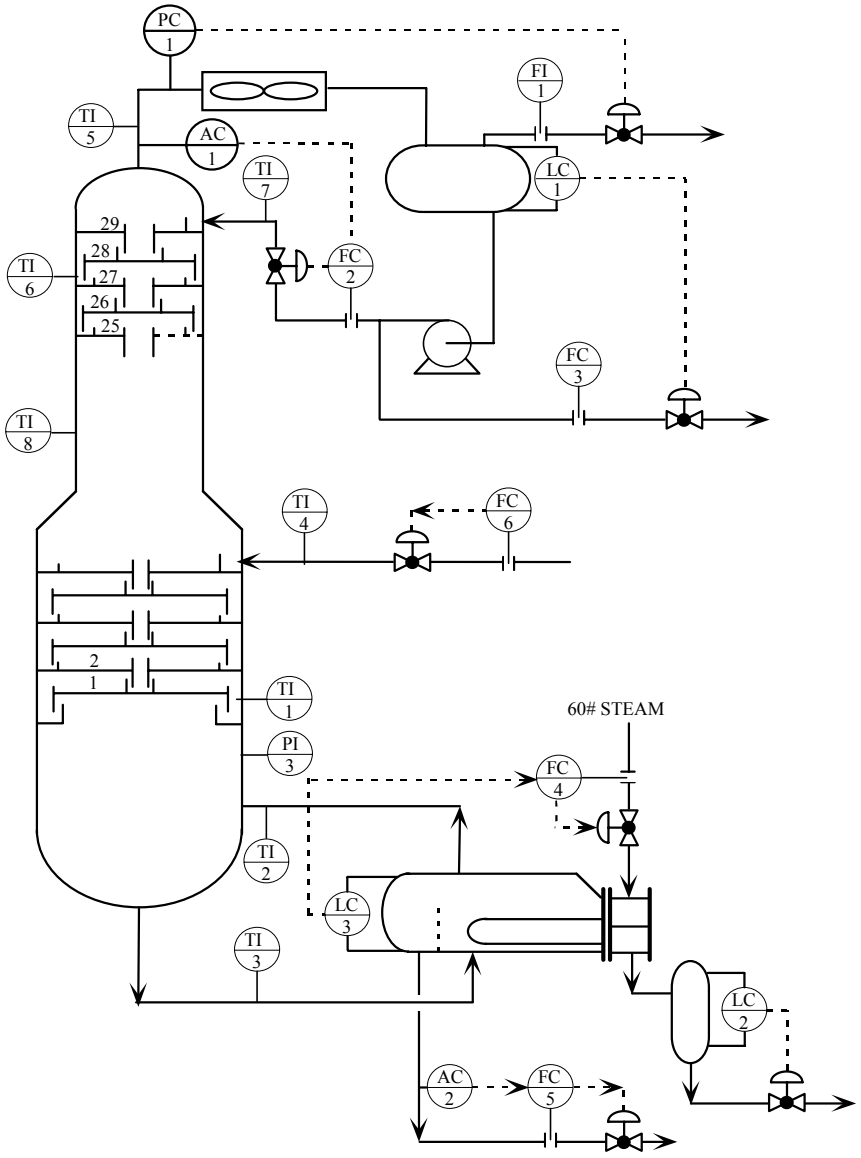

Figure 7.5: De-ethanizer Column Process Flow Diagram.

Initially, one hidden layer with five neurons is built (additional neurons and/or layers can be added if necessary) and all weights are randomly initialized to small numbers. Next, training parameters are defined. These parameters include the following: maximum number of training iterations and acceptable error between desired and predicted values.

The neural network program using back propagation training algorithm starts training and through this process it will look for the specified error on a multidimensional surface. By selecting the minimum error to be a very small number (10^{-3} for example) the program will end up in one of the following states:

1. Minimum error goal is matched before exceeding the limit on maximum allowed iterations. In this case, the objective of the training is successfully met.
2. Program cannot achieve this minimum error but, in the process, it locates the global minimum (optimum solution). In this case, the number of hidden neurons and/or the number of hidden layers can be increased to achieve the desired minimum error.
3. Training diverges. The error increases as the training process continues. (Training data sets are not valid.) In this case it is necessary to construct valid data sets.

7.6 SIMULATION RESULTS AND DISCUSSIONS

As discussed earlier, the objective of the proposed work is to eliminate the dependency on laboratory and/or on-line sample analyzers for sampling of product qualities. The goal can be achieved by the construction of neural networks to predict those particular product qualities to meet the more stringent market specifications. In doing so, the neural network model, from a practical viewpoint, should adhere to two constraints: The optimization of process control and the reduction on the cost of maintenance and operations, which would ultimately result in an increase in profit [3].

First, the neural network model accuracy of prediction should be consistent and within the defined acceptable tolerance of the desired product quality it is set to predict. It is highly crucial to have a neural network that provides accurate predictions. It is a plant requirement to have the neural network predicted output fed as one of the inputs to a multivariable controller. This will provide the controller with the knowledge of the final product quality, and how close to or far from the desired set point it is. With the aid of this knowledge, the controller will act promptly to keep the process in its targeted path, thus eliminating any off-specs product from taking place.

Secondly, it is a requirement to have the neural network running on-line with fast execution time during both training and prediction phases. The multivariable controller is gathering information about the process and at the same time it is looking at the neural network to provide its prediction. The controller will perform its tight control actions as long as the neural network prediction is made available to the controller at the right moment, not a couple of minutes late. Also, operational objectives often change to meet market needs and in doing so the desired process set points have to change as well to provide the desired product specifications. Retraining the neural network on the new sets of process variables and desired product properties is inevitable. The faster the neural

network program predicts after retraining, the faster it provides its output to the controller [3], [9].

7.6.1 Naphtha 95% Cut Point

Modeling of the naphtha 95% cut point property was carried out using a back propagation neural network algorithm. Various configurations, in terms of the number of hidden layers and the number of hidden neurons, have been tested. For the application presented here, two-layer networks consisting of a single hidden layer and an output layer have proved to be adequate. Although a three-layer network is theoretically capable of modeling more general and arbitrary functions than a two-layer network [17], the naphtha 95% cut point data used in training and verification modes were sufficiently well behaved that three-layer networks did not perform better than the ones consisting of two layers.

To demonstrate the modeling capability of a back propagation network, 85 data sets were analyzed. Each data set consisted of 33 process variables as inputs to the model and one product quality (naphtha 95% cut point) as an output. A total of 70 data sets were used in the training phase and 15 data sets were used in the verification phase. Table 7.4 summarizes the simulation results.

For the first case, a single hidden layer consisting of five neurons was utilized. The model could not achieve the desired error goal of 0.01 after performing 10,000 iterations, which was the maximum allowed number of iterations. A maximum error of 7.84°F at training phase was obtained. . In the verification phase, a maximum error of 11.59°F. was detected. Figures 7.6 and 7.7 shows the results of the training and verification phases, respectively.

For further investigation, a first-momentum term was added to the back propagation algorithm. However, the model still could not achieve the desired error goal after 10,000 iterations as shown on the table. Finally, with the same model as above, an adaptive learning rate was added, and the neural network model achieved the desired sum squared error goal of 0.01 in 3180 iterations.

Next, the number of hidden neurons was increased. Table 7.5 also shows the training and verification results using eight neurons in the hidden layer. The model was able to achieve an acceptable error in the training phase of 1.35°F but failed to achieve comparable results in the verification phase where the maximum absolute error was 6.82°F. Further increase in the number of hidden neurons only improved the results in the training phase. The verification phase continued to show error values too significant to be accepted for good prediction of the Naphtha 95% cut point property.

The next step in the simulation was to increase the number of hidden layers. Two hidden layers were selected and the number of neurons in each layer was varied. Figures 7.8 and 7.9 show the training and verification results using eight neurons in the first hidden layer and four neurons in the second hidden layer. The result shows slight improvement in the verification phase but more accurate prediction is still required. It can be noticed that in the training phase the models performed well, however, in the verification phase all the tested models could

not predict with enough accuracy. It was suspected that the neural network models were memorizing the relationship between the inputs and the output since they were trying to adhere to a very small error goal in the training phase.

Table 7.5: Initial Simulation Results for Naphtha 95% Cut Point

Hidden Neurons	Training Phase				Verification Phase	
	Error goal	Iterations	Final SSE	Max. Error °F	Final SSE	Max. Error °F
5 BP	0.01	10000	0.045	7.84	0.45	11.59
5M	0.01	10000	0.031	1.57	0.22	7.92
5	0.01	3180	0.01	2.17	0.28	5.49
8	0.01	4563	0.01	1.35	0.25	3.29
10	0.01	2088	0.01	1.83	0.27	7.95
8-4	0.01	4302	0.01	1.33	0.14	4.81

It is important to prevent the neural network model from memorizing the input/output relationship. A neural network with enough hidden neurons given enough iterations and a very small error goal will actually memorize a given relationship between model inputs and outputs. In other words, a network memorizes relationships between outputs and inputs when the model building points are allowed to conform to a degree much less than lab repeatability. It means that an acceptable sum squared error goal in the training phase should generate a degree of accuracy very close to lab repeatability. A typical value used for lab repeatability for the naphtha 95% cut point is 3.6°F. If one insists on achieving a degree of accuracy greater than lab repeatability, the network memorizes the relationship during the training process; this is known as overfitting. When overfitting occurs, each data point during the training is fit perfectly but the network is not able to predict with the same accuracy during the verification phase. A two-layer network with 12 neurons in the hidden layer was trained with an error goal of 0.05 (Table 7.5) to yield a maximum error of 1.7 °F in the training phase. The maximum error in the verification phase was 5.79°F. The network could not generalize. It memorized the relation between inputs and outputs in the training phase and did not follow the general trend of the relation between inputs and targets.

Table 7.5: Simulation Results for Naphtha 95% Cut Point

Hidden Neurons	Training Phase				Verification Phase	
	Error goal	Iterations	Final SSE	Max. Error °F	Final SSE	Max. Error °F
5	0.1	374	0.1	3.21	0.091	4.17
	0.3	196	0.3	6.61	0.432	13.76
8	0.1	238	0.1	2.87	0.097	3.29
	0.05	681	0.05	1.8	0.111	3.99

	0.01	5686	0.01	1.37	0.161	4.7
12	0.1	385	0.1	3.18	0.117	4.87
	0.05	1237	0.05	1.7	0.145	5.79
5 - 2	0.1	269	0.1	3.03	0.098	5
	0.2	211	0.2	3.81	0.127	5.45

Table 7.5 shows a summary of the simulation results. The best model architecture (in terms of better prediction in both training and verification modes) consists of eight neurons in one hidden layer. Both hidden and output layers use sigmoidal activation functions as the nonlinear element for their neurons. The model is trained to achieve a sum squared error goal of 0.1. The sum squared error goal in the verification mode is 0.097.

In this application it is important that the neural network output is equal to or less than the acceptable error. As mentioned earlier, the acceptable error value is based on lab repeatability. For the naphtha 95% cut point, this value is 3.6°F. Further data analysis is performed to look at the absolute error in each data set in both the training and verification modes. The maximum absolute error in the training data sets is 2.87°F, whereas in the verification mode the maximum absolute error is 3.29°F.

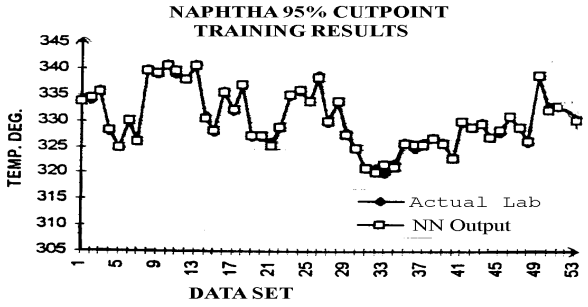

Figure 7.6: L_{inf} Error Norm in the Training Phase.
(Hidden Layer Neurons: 5, Error Goal=0.01, Max. Error=7.84°F)

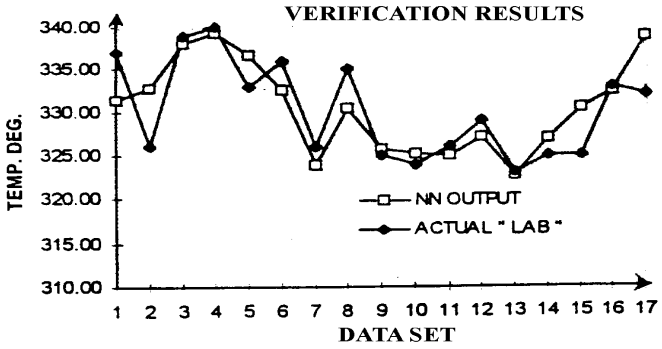

Figure 7.7: L*inf* Error Norm in the Verification Phase.
(Hidden Layer Neurons = 5, Error Goal = 0. 01, Max. Error = 7.84°F)

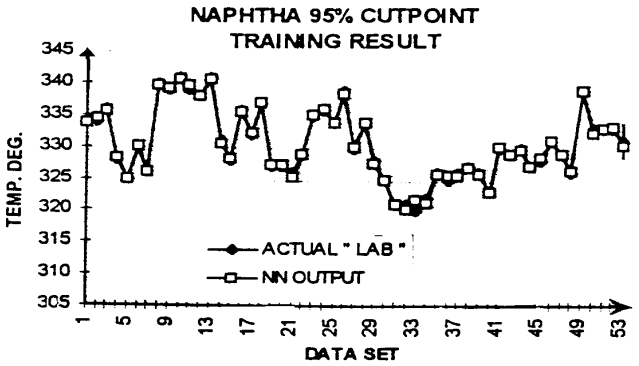

Figure 7.8: L_{inf} Error Norm in the Training Phase.
(Hidden Layers Neurons = 8,4; Error Goal = 0.01, Max. Error =1.33°F)

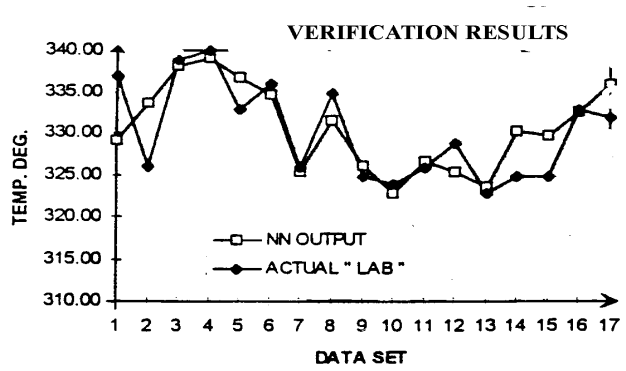

Figure 7.8: L_{inf} Error Norm in Verification Phase.
(Hidden Layers Neurons = 8,4; Error Goal = 0.01, Max. Error = 4.81°F)

7.6.2 Naphtha Reid Vapor Pressure

To demonstrate the modeling capability of a back propagation algorithm for RVP prediction parameters, 83 data sets were analyzed. Each data set consisted of seven process variables (Table 7.6) as inputs to the model and one product quality (Reid vapor pressure) as an output. A total of fifty five data sets were used in the training phase and 28 data sets are used in the verification phase.

Table 7.6: Inputs to the RVP Neural Network Model

Process Variables
FEED FLOW RATE
REFLUX FLOW RATE
OVERHEAD PRESSURE
FEED TEMPERATURE
REBOILER RETURN TEMP.
TRAY 2 TEMPERATURE
COLUMN TOP TEMPERATURE

The best model architecture (in terms of better prediction in both training and verification modes) consisted of five neurons in one hidden layer. Both hidden and output layers use sigmoidal activation functions as the nonlinear element for their neurons. The model is trained to achieve a sum squared error goal of 0.1. The sum squared error goal in the verification mode is 0.097. The maximum absolute error in the training data sets is 0.21 psi, whereas in the verification mode the maximum absolute error is 0.48 psi. The detail work can be found in reference [3].

7.7 CONCLUSIONS

In this chapter, various neural network architectures are proposed for the prediction of product quality of an oil refinery. The important parameters involved in acquiring valid data sets are considered. Close attention is paid to the proper selection of the input data. Finally, two product quality properties, namely, naphtha 95% cut point and naphtha Reid vapor pressure, were successfully modeled using neural network.

After the generation of the neural network models, the central processing computer system of an oil refinery may use them on-line. Using the NN model on-line is straightforward except for one point of caution. The network was trained within a specific range for the different process variables and the lab data. It is important to realize that while neural network models are excellent interpolators, they can be bad extrapolators due to the non-linearity of the correlation generated. It is, therefore, important to check process parameters used in the prediction and to make sure that these parameters fall within the range that was used to create the model. If parameters fall out of range, then the predicted lab value is questionable. Lab data collected while process parameters are outside the range can be used to further expand the window of operation of the neural network model. As the variability in plant operation increases, and the

network window expands, the generation models can become more and more reliable.

REFERENCES

1. Antsaklis, P. J. and Passino, K.M. (eds.), *An Introduction to Intelligent and Autonomous Control*, Kluwer Academic Publishers, Norwell, MA, 1993.
2. *Ras Tanura Refinary Facilities Manual*, Ras Tanura Refinary: Refining Division, Section 3, 2nd ed., 1995.
3. Bawazeer, K. H., Prediction of Crude Oil Product Quality Parameters Using Neural Networks, MS Thesis, Florida Atlantic University, Boca Raton, FL, August, 1996.
4. Bawazeer, K.H. and Zilouchian, A., Prediction of Crude Oil Production Quality Parameters Using Neural Networks, *Proc. of 1997 IEEE Int. Conf. on Neural Networks*., New Orleans, 1997.
5. Borman, S., Neural Network Applications in Chemistry Begin to Appear, *Chemical Eng. News*, Vol. 67, No. 17, 24-29, 1989.
6. Parlos, A. G., Chong, K.T., and Atiya, A.F., Application of Recurrent Neural Multilayer Perceptron in Modeling Complex Process Dynamic, *IEEE Trans. on Neural Networks*, Vol. 5, No. 2, March, 1994.
7. Nekovie, R., and Sun,Y., Back Propagation Network and its Configuration for Blood Vessel Detection in Angiograms, *IEEE Trans. on Neural Networks*, Vol. 6, No. 1, 1995.
8. Berkan, R.C., Upadhyaya, B., Tsoukalas, L., Kisner, R., and Bywater, R. Advanced Automation Concepts for Large-Scaled Systems, *IEEE Control Syst. Mag.*, Vol. 11, No. 6, 4–13, Oct., 1991.
9. Draeger, A., Engell, S., and Ranke, H., Model Predictive Control Using Neural Networks, *IEEE Control Mag.*. Vol. 15, No.5, 61-67, 1995.
10. Ray, W., Polymerization Reactor Control, *IEEE Control Syst. Mag.*, Vol. 6, No. 4, 3–9, August, 1986.
11. Bhat, N., Minderman, P., McAvoy, T., and Wang, N., Modeling Chemical Process Systems via Neural Network Computation, *IEEE Control Syst. Mag.*, Vol. 10, No.3, 24–31, April, 1990.
12. Rosenblatt, A., *Principles of Neurodynamics*, Spartan Press, Washington, DC, 1961.
13. Fausett, L., *Fundamentals of Neural Networks*, Prentice-Hall, Englewood Cliffs, NJ, 1994.
14. Demuth, H. and Beale, M., *Neural Network Toolbox for Use with MATLAB* , the Math Works Inc. , Natick, MA, 1998.
15. Miller, W.T., Suton, R., and Werbos, P. (eds.), *Neural Networks for Control*, MIT Press, MA, 1990.
16. Lippmann, R.P., An Introduction to Computing with Neural Network; *IEEE Acoustic, Speech, and Signal Proc. Mag,* 4–22, April, 1987.

17. Kosko, B., *Neural Networks and Fuzzy Systems,* Prentice-Hall, Englewood Cliffs, NJ, 1991.