NEAR EAST UNIVERSITY

GRADUATE SCHOOL OF APPLIED AND SOCIAL SCIENCES

ANN BASED PRODUCT QUALITY PREDICTION FOR CRUDE DISTILLATION UNIT

Filiz Alshanableh

Master Thesis

Department of Electrical and Electronic Engineering

Nicosia-2005

Filiz Alshanableh: ANN Based Product Quality Prediction for Crude Distillation Unit

Approval of the Graduate School of Applied and Social Sciences

Prof. Dr. Fahreddin Sadıkoğlu
Director

We certify this thesis is satisfactory for the award of the Degree of Master of Science in Electrical & Electronic Engineering

Examining Committee in charge:

Prof. Dr. Fahreddin Sadıkoğlu, Chairman of Committee, Electrical and Electronic Engineering Department, NEU

Assist. Prof. Dr. Kadri Bürüncük, Member, Electrical and Electronic Engineering Department, NEU

Assist. Prof. Dr. Erdal Onurhan, Member, Mechanical Engineering Department, NEU

Assoc. Prof. Dr. Rahib Abiyev, Supervisor, Computer Engineering Department, NEU

ACKNOWLEDGEMENT

First, I would like to thank my supervisor Assoc. Prof. Dr. Rahib Abiyev for his invaluable support, advice and courage that he gave to me to continue my thesis.

I would also like to express my gratitude to Near East University for providing me such an environment that made this work possible.

I thank my mum, dad and sisters for their belief in me in all my life attempts.

Finally, I would like to thank the most important person of my life, to my dear husband Tayseer for his life time courage, support, advice and belief in myself.

I dedicate this thesis to my beloved children Muhammed, Nur and Yasemin.

ABSTRACT

In industry, under real time conditions, describing the state of production in a finite time interval often requires processing of great volume of information. This requires developing a system that would process the coming information in parallel and high level of reliability. One of the approaches that meet the above requirements is Neural Networks.

In this thesis the development of quality prediction system for Crude Distillation Unit (CDU) products is considered. The analysis and technological description of CDU is given. Quality of the products depends on many parameters. The main technological parameters that influence to the output products of CDU have been observed. Artificial Neural Network is used to predict product quality in the CDU technological process.

The mathematical models of Neural Network and its learning algorithm are given. Using Neural Network structure the development of the quality prediction is carried out. For prediction the Naphtha 95 % Cut Point property is chosen.

Using statistical data taken from technological process and implementing the back propagation learning algorithm, product quality prediction for naphtha 95 % Cut Point has been performed. Development of the system is realized using Neuroshell software package and NNinExcell software package and results of simulation with both packages are analyzed.

CONTENTS

INTRODUCTION

In response to demand 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 reduce 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 the specific range of operation. However, the existence of highly nonlinear relationships between the process variables (inputs) and 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. The implementation of intelligent control technology based on soft computing methodologies such as neural network (NN) can remarkably enhance the regulatory and advance control capabilities of various industrial processes such as oil refineries.

Presently, in the majority of oil refineries (such as Tüpraş Refinery, in İzmit, Turkey), 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 specification within the acceptable tolerance, the product has to be reprocessed to meet the required specification. This process is costly in terms of time and money. In order to solve 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 thesis, the utilization of artificial neural network (ANN) technology for the inferential analysis of crude fractionation section of Tüpraş Refinery, in İzmit , Turkey is presented. The implementation of several neural network models using back propagation algorithm based on collection of real-time data for a four months operation of a plant is presented. The proposed neural network architecture can accurately predict various properties associated with crude oil production. The result of the proposed work can ultimately enhance the on-line prediction of crude oil product quality parameters for the crude distillation (fractionation) processes of various oil refineries.

The thesis consists of four chapters and a conclusion. First two chapters give an introduction about the background of this work; technological process described focusing on Crude Distillation Unit and Neural Networks learning and the last two chapters explain the work done.

In Chapter 1, description of the refinery process including basics of crude oil as raw material of refinery process and major refinery products are presented. Since this thesis will be focused on the process of the Crude Distillation Unit, which is the starting point for all refinery operations, complete process description of the Crude Distillation Unit will be given.

In Chapter 2, an introduction about the neural networks, development of neural networks, structure of neural networks that is included biological neural networks, artificial models and components of artificial neuron are presented. Also classification of neural network learning as supervised and unsupervised will be described. Finally, back propagation and its algorithm will be explained in details.

In Chapter 3, development of neural network system of product quality prediction is described. A structure of neural network system to predict product quality will be presented. Selection of process variables that have influence to product quality is determined. The main steps for development of neural network system to predict naphtha cut point will be explained in details.

In Chapter 4, the neural network learning structure and the training procedures as well as the results of the modelling for naphtha 95 % cut point will be analyzed.

In the Conclusion, the results of this work will be explained.

1. TECHNOLOGICAL PROCESS DESCRIPTION

1.1 Overview

This chapter gives description of the refinery process including basics of crude oil as raw material of refinery process and major refinery products. Since this thesis will be focused on process of the Crude Distillation Unit that is the starting point for all refinery operations, complete process description of the Crude Distillation Unit will be given.

1.2 Description of the Refinery Process

The petroleum industry began with the successful drilling of the first commercial oil well in 1859, and the opening of the first refinery two years later to process the crude into kerosene. The evolution of petroleum refining from simple distillation to today's sophisticated processes has created a need for technological improvement. To those unfamiliar with the industry, petroleum refineries may appear to be complex and confusing places. Refining is the processing of one complex mixture of hydrocarbons into a number of other complex mixtures of hydrocarbons. Petroleum refining has evolved continuously in response to changing consumer demand for better and different products. The original requirement was to produce kerosene as a cheaper and better source of light than whale oil. The development of the internal combustion engine led to the production of gasoline and diesel fuels. The evolution of the airplane created a need first for high-octane aviation gasoline and then for jet fuel, a sophisticated form of the original product, kerosene. Present-day refineries produce a variety of products including many required as feedstock for the petrochemical industry [1]. Although here description of refinery process is given, attention will be focused on Crude Distillation Unit operation.

1.3 Crude Oil

1.3.1 Basics of Crude Oil

Crude oils are complex mixtures containing many different hydrocarbon compounds that vary in appearance and composition from one oil field to another. Crude oils range in consistency from water to tar-like solids, and in color from clear to black. An "average" crude oil contains about 84% carbon, 14% hydrogen, 1%-3% sulfur, and less than 1% each of nitrogen, oxygen, metals, and salts. Crude oils are generally classified as paraffinic, naphthenic, or aromatic, based on the predominant proportion of similar hydrocarbon molecules. Mixed-base crude has varying amounts of each type of hydrocarbon. Refinery crude base stocks usually consist of mixtures of two or more different crude oils.

Crude oils are defined in terms of API (American Petroleum Institute) gravity. The higher the API gravity, the lighter the crude. For example, light crude oils have high API gravities and low specific gravities. Crude oils with low carbon, high hydrogen, and high API gravity are usually rich in paraffins and tend to yield greater proportions of gasoline and light petroleum products; those with high carbon, low hydrogen, and low API gravities are usually rich in aromatics.

Crude oils that contain appreciable quantities of hydrogen sulfide or other reactive sulfur compounds are called "sour." Those with less sulfur are called "sweet." Some exceptions to this rule are West Texas crude, which are always considered "sour" regardless of their H_2S content, and Arabian high-sulfur crude, which are not considered "sour" because their sulfur compounds are not highly reactive [1].

1.3.2 Major Refinery Products

- **Gasoline:** The most important refinery product is motor gasoline, a blend of hydrocarbons with boiling ranges from ambient temperatures to about 400 °F. The important qualities for gasoline are octane number (antiknock), volatility (starting and vapor lock), and vapor pressure (environmental control). Additives are often used to enhance performance and provide protection against oxidation and rust formation.

- **Kerosene:** Kerosene is a refined middle-distillate petroleum product that finds considerable use as a jet fuel and around the world in cooking and space heating. When used as a jet fuel, some of the critical qualities are freeze point, flash point, and smoke point. Commercial jet fuel has a boiling range of about 375°-525 °F, and military jet fuel 130°-550 °F. Kerosene, with less-critical specifications, is used for lighting, heating, solvents, and blending into diesel fuel.

- **Liquified Petroleum Gas (LPG):** LPG, which consists principally of propane and butane, is produced for use as fuel and is an intermediate material in the manufacture of petrochemicals. The important specifications for proper performance include vapor pressure and control of contaminants.

- **Distillate Fuels:** Diesel fuels and domestic heating oils have boiling ranges of about 400°-700 °F. The desirable qualities required for distillate fuels include controlled flash and pour points, clean burning, no deposit formation in storage tanks, and a proper diesel fuel cetane rating for good starting and combustion.

- **Residual Fuels:** Many marine vessels, power plants, commercial buildings and industrial facilities use residual fuels or combinations of residual and distillate fuels for heating and processing. The two most critical specifications of residual fuels are viscosity and low sulfur content for environmental control.

- **Coke and Asphalt:** Coke is almost pure carbon with a variety of uses from electrodes to charcoal briquettes. Asphalt, used for roads and roofing materials, must be inert to most chemicals and weather conditions.

- **Solvents:** A variety of products, whose boiling points and hydrocarbon composition are closely controlled, are produced for use as solvents. These include benzene, toluene, and xylene.

- **Petrochemicals:** Many products derived from crude oil refining, such as ethylene, propylene, butylenes, and isobutylene, are primarily intended for use as petrochemical feedstock in the production of plastics, synthetic fibers, synthetic rubbers, and other products.

- **Lubricants:** Special refining processes produce lubricating oil base stocks. Additives such as demulsifiers, antioxidants, and viscosity improvers are blended into the base stocks to provide the characteristics required for motor oils, industrial greases, lubricants, and cutting oils. The most critical quality for lubricating-oil base stock is a high viscosity index, which provides for greater consistency under varying temperatures [1].

1.4 Petroleum Refining Process

Petroleum refining begins with the distillation, or fractionation, of crude oils into separate hydrocarbon groups. The resultant products are directly related to the characteristics of the crude processed. Most distillation products are further converted into more usable products by changing the size and structure of the hydrocarbon molecules through cracking, reforming, and other conversion processes as discussed in this chapter. These converted products are then subjected to various treatment and separation processes such as extraction, hydro treating, and sweetening to remove undesirable constituents and improve product quality. Integrated refineries incorporate fractionation, conversion, treatment, and blending operations and may also include petrochemical processing.

1.4.1 Refining Operations

Petroleum refining processes and operations can be separated into five basic areas:

- Fractionation (distillation) is the separation of crude oil in atmospheric and vacuum distillation towers into groups of hydrocarbon compounds of differing boiling-point ranges called "fractions" or "cuts."

- Conversion processes change the size and/or structure of hydrocarbon molecules. These processes include:
Decomposition (dividing) by thermal and catalytic cracking;
Unification (combining) through alkylation and polymerization; and
Alteration (rearranging) with isomerization and catalytic reforming.

- Treatment processes are intended to prepare hydrocarbon streams for additional processing and to prepare finished products. Treatment may include the removal or separation of aromatics and naphthenes as well as impurities and undesirable contaminants. Treatment may involve chemical or physical separation such as dissolving, absorption, or precipitation using a variety and combination of processes including desalting, drying, hydrodesulfurizing, solvent refining, sweetening, solvent extraction, and solvent dewaxing.

- Formulating and Blending is the process of mixing and combining hydrocarbon fractions, additives, and other components to produce finished products with specific performance properties.

- Other Refining Operations include: light-ends recovery; sour-water stripping; solid waste and wastewater treatment; process-water treatment and cooling; storage and handling; product movement; hydrogen production; acid and tail-gas treatment; and sulfur recovery.

Auxiliary operations and facilities include: steam and power generation; process and fire water systems; flares and relief systems; furnaces and heaters; pumps and valves; supply of steam, air, nitrogen, and other plant gases; alarms and sensors; noise and pollution controls; sampling, testing, and inspecting; and laboratory, control room, maintenance, and administrative facilities [1].

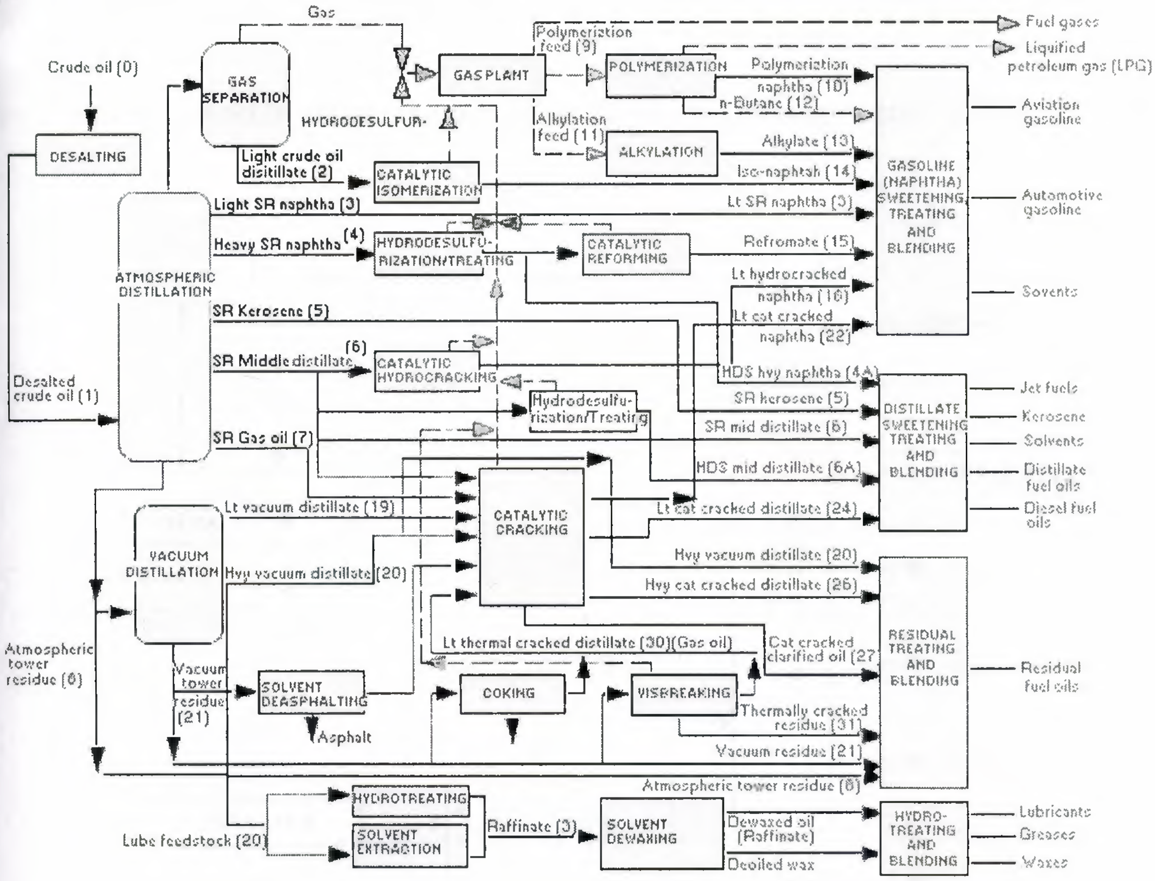


Figure 1.1 Refinery process chart

1.5 Crude Oil Distillation Process

1.5.1 Description

The crude distillation unit (CDU) is the starting point for all refinery operations. The first step in the refining process is the separation of crude oil into various fractions or straight-run cuts by distillation in atmospheric and vacuum towers. The separation of crude oil into raw products is accomplished in the crude unit by fractional distillation in fractionating columns, based on their distillation range. The process does not involve any chemical changes. The main fractions or "cuts" obtained have specific boiling-point ranges and can be classified in order of decreasing volatility into gases, light distillates, middle distillates, gas oils, and residuum.

1.5.2 Atmospheric Distillation Tower

A schematic representation of the crude oil and product flow is presented in Fig 1.2.

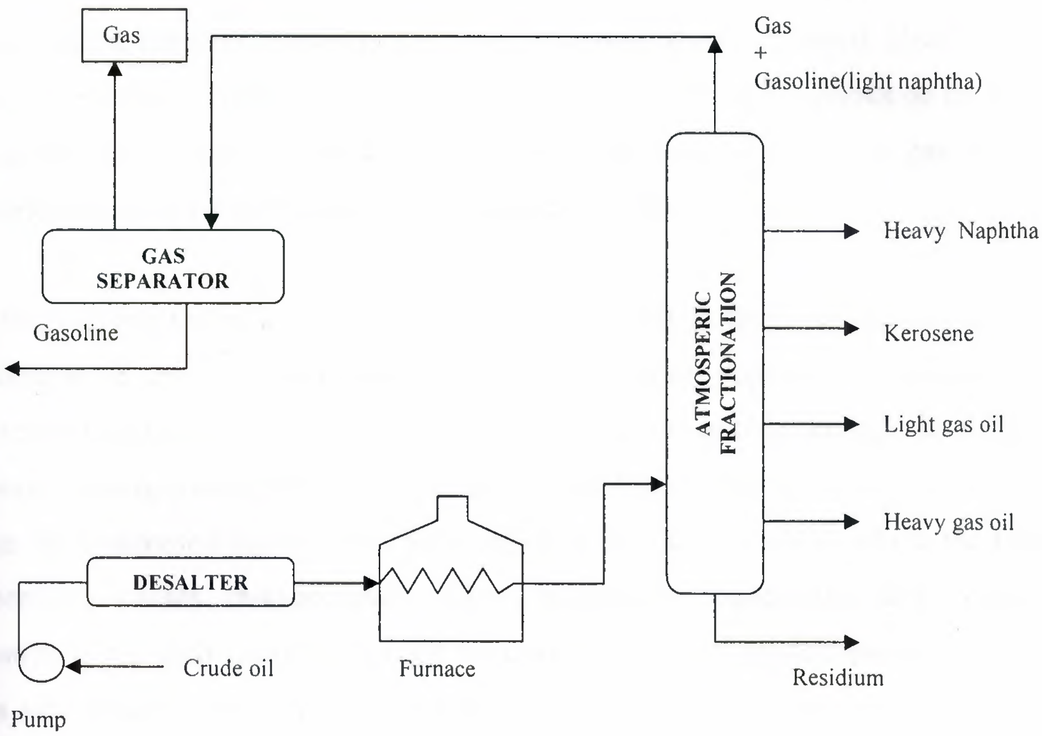


Figure 1.2 Atmospheric Distillation Unit

The crude feed pump, located near the crude storage tanks, supplies the feed to the unit. The feed to the unit is passed through a desalter where the chlorides of calcium, magnesium and sodium are removed. These salts form corrosive acids during processing and therefore are detrimental to process equipments. By injecting water to the crude oil stream these salts are dissolved in the water and the solution is separated from the crude by means of an electrostatic separator in a large vessel. The electrically charged grids coalesces the water and aids separation from the crude. After desalting the crude is heated through a series of heat exchangers and then by a furnace to a temperature of 700°F and admitted to the flash zone of the atmospheric distillation tower [2].

The desalted crude feedstock is preheated using recovered process heat. The feedstock then flows to a direct-fired crude charge heater where it is fed into the vertical distillation column just above the bottom, at pressures slightly above atmospheric and at temperatures ranging from 650° to 700° F (heating crude oil above these temperatures may cause undesirable thermal cracking). All but the heaviest fractions flash into vapour. As the hot vapour rises in the tower, its temperature is reduced. Heavy fuel oil or asphalt residue is taken from the bottom. At successively higher points on the tower, the various major products including lubricating oil, heating oil, kerosene, gasoline, and uncondensed gases (which condense at lower temperatures) are drawn off.

The fractionating tower, a steel cylinder about 35 m high, contains horizontal steel trays for separating and collecting the liquids. At each tray, vapours from below enter perforations and bubble caps. They permit the vapours to bubble through the liquid on the tray, causing some condensation at the temperature of that tray. An overflow pipe drains the condensed liquids from each tray back to the tray below, where the higher temperature causes re-evaporation. The evaporation, condensing, and scrubbing operation is repeated many times until the desired degree of product purity is reached. Then side streams from certain trays are taken off to obtain the desired fractions. Products ranging from uncondensed fixed gases at the top to heavy fuel oils at the bottom can be taken continuously from a fractionating tower. Steam is often used in towers to lower the vapour pressure and create a partial vacuum. The distillation process separates the major constituents of crude oil into so-called straight-run products. Sometimes crude oil is "topped" by distilling off only the lighter fractions, leaving a heavy residue that is often distilled further under high vacuum [1].

Four fractions are separated in the atmospheric tower. The overhead vapours are condensed in a two stage system. The condensed liquid from the first stage is used as reflux to the tower. The second stage liquid together with the compressed and condensed vapours from the second stage is collected in the stabilizer feed accumulator. The liquid in the stabilizer feed accumulator is the feed to the Vapour Recovery unit. The uncondensed vapours from the stabilizer feed accumulator is routed to fuel gas system after removal of H_2S in the sulphur plant. The other three products separated are heavy naphtha, kerosene and diesel. The heavy naphtha is drawn from the next tray and is steam stripped to improve flash. The majority of this product is line blended with diesel from HSD (High Speed Diesel oil) desulphurisation unit and raw diesel to make finished high speed diesel oil. A small amount of the heavy naphtha is sent to Merox treater. This treater oxidises mercaptans to disulphides thereby eliminating the unpleasant odour. Kerosene drawn from the lower tray is steam stripped and is charged hot to kerosene hydro-desulphuriser plant. When this unit is shut down, kerosene is cooled and sent to intermediate storage tank through the kerosene product cooler.

Diesel oil is drawn from the next plate. Approximately 50% of the diesel oil is routed to HSD desulphurisation unit after heat exchange with crude and the balance is cooled and blended with the desulphurised diesel oil to produce HSD product. The stripped overhead liquid streams from kerosene hydrode-sulphuriser, HSD desulphuriser and lube oil hydrofinisher are sent to the atmospheric distillation tower after separating the water in a dewatering drum.

The hot reduced crude from the bottom of atmospheric distillation tower is further fractionated in the two stage vacuum distillation section. The vacuum maintained in these fractionators makes it possible to fractionate the reduced crude at much lower temperatures. But for this vacuum, the higher temperatures required to fractionate reduced crude will result in cracking of the products.

The reduced crude from atmospheric tower bottoms is further heated in presence of steam in the first stage vacuum heater and introduced into the first stage vacuum tower. Three side-stream products spindle oil, light neutral and intermediate neutral and an overhead product—gas oil are separated in the first stage vacuum tower. Spindle oil, light neutral and intermediate neutral are sent to the Lube Oil Extraction plants as feed stock or to storage. The bottoms product from first stage vacuum tower is reheated along with steam and fractionated to yield heavy neutral stream. Flash zone vapours of the second stage vacuum tower pass through a demister pad to prevent entrainment of asphaltenes into the heavy neutral stream [2].

1.6 Summary

Since in this thesis neural network system is applied to predict product quality on process of the Crude Distillation Unit that is the starting point for all refinery operations, complete process description of the Crude Distillation Unit was given in this chapter.

2. NEURAL NETWORKS

2.1 Overview

This chapter presents an introduction about the neural networks, structure of neural networks including artificial models and components of artificial neuron. Also classification of neural network learning as supervised and unsupervised will be described. Finally, back propagation learning and its algorithm will be explained in details.

2.2 Introduction to Neural Networks

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well. Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions. Other advantages include:

- Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.

- Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time.
- Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.

- Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

2.3 An Artificial Neuron

The fundamental processing element of a neural network is a neuron. This building block of human awareness encompasses a few general capabilities. Basically, a biological neuron receives inputs from other sources, combines them in some way, performs a generally nonlinear operation on the result, and then outputs the final result.

Biological neurons are structurally more complex than the existing artificial neurons that are built into today's artificial neural networks. As biology provides a better understanding of neurons, and as technology advances, network designers can continue to improve their systems by building upon man's understanding of the biological brain. But currently, the goal of artificial neural networks is not the grandiose recreation of the brain. On the contrary, neural network researchers are seeking an understanding of nature's capabilities for which people can engineer solutions to problems that have not been solved by traditional computing. To do this, the basic unit of neural networks, the artificial neurons, simulates the four basic functions of natural neurons.

In Figure 2.1, various inputs to the network are represented by the mathematical symbol, x_n. Each of these inputs are multiplied by a connection weight. These weights are represented by w_{in}. In the simplest case, these products are simply summed, fed through a transfer function to generate a result, and then output. This process lends itself to physical implementation on a large scale in a small package. This electronic implementation is still possible with other network structures which utilize different summing functions as well as different transfer functions.

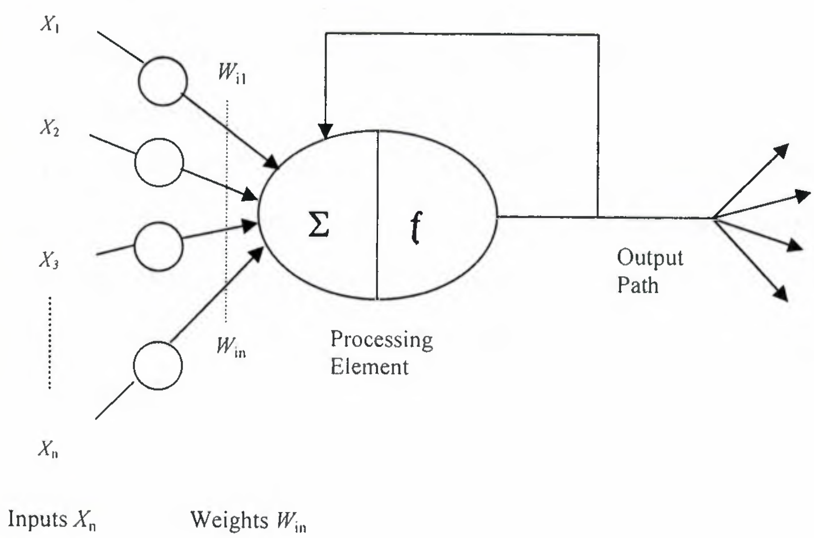


Figure 2.1 A Basic Artificial Neuron.

2.3.1 Major Components of an Artificial Neuron

This section describes the seven major components which make up an artificial neuron [4]. These components are valid whether the neuron is used for input, output, or is in one of the hidden layers.

2.3.1.1. Weighting Factors

A neuron usually receives many simultaneous inputs. Each input has its own relative weight which gives the input the impact that it needs on the processing element's summation function. These weights perform the same type of function as do the varying synaptic strengths of biological neurons. In both cases, some inputs are made more important than others so that they have a greater effect on the processing element as they combine to produce a neural response. Weights are adaptive coefficients within the network that determine the intensity of the input signal as registered by the artificial neuron. They are a measure of an input's connection strength. These strengths can be modified in response to various training sets and according to a network's specific topology or through its learning rules.

2.3.1.2. Summation Function

The first step in a processing element's operation is to compute the weighted sum of all of the inputs. Mathematically, the inputs and the corresponding weights are vectors which can be represented as $(x_1, x_2, \ldots x_n)$ and $(w_1, w_2 \ldots w_n)$. The total input signal is the dot, or inner, product of these two vectors. This simplistic summation function is found by multiplying each component of the x vector by the corresponding component of the w vector and then adding up all the products. $\text{Input}_1 = x_1 * w_1$, $\text{input}_2 = x_2 * w_2$, etc., are added as $\text{input}_1 + \text{input}_2 + \ldots + \text{input}_n$. The result is a single number, not a multi-element vector.

The summation function can be more complex than just the simple input and weight sum of products. The input and weighting coefficients can be combined in many different ways before passing on to the transfer function. In addition to a simple product summing, the summation function can select the minimum, maximum, majority, product, or several normalizing algorithms. The specific algorithm for combining neural inputs is determined by the chosen network architecture and paradigm.

Some summation functions have an additional process applied to the result before it is passed on to the transfer function. This process is sometimes called the activation function. The purpose of utilizing an activation function is to allow the summation output to vary with respect to time. Activation functions currently are pretty much confined to research. Most of the current network implementations use an "identity" activation function, which is equivalent to not having one. Additionally, such a function is likely to be a component of the network as a whole rather than of each individual processing element component.

2.3.1.3. Transfer Function

The result of the summation function, almost always the weighted sum, is transformed to a working output through an algorithmic process known as the transfer function. In the transfer function the summation total can be compared with some threshold to determine the neural output. If the sum is greater than the threshold value, the processing element generates a signal. If the sum of the input and weight products is less than the threshold, no signal (or some inhibitory signal) is generated. Both types of response are significant. The threshold, or transfer function, is generally non-linear. Linear (straight-line) functions are limited because the output is simply proportional to the input. Linear functions are not very useful.

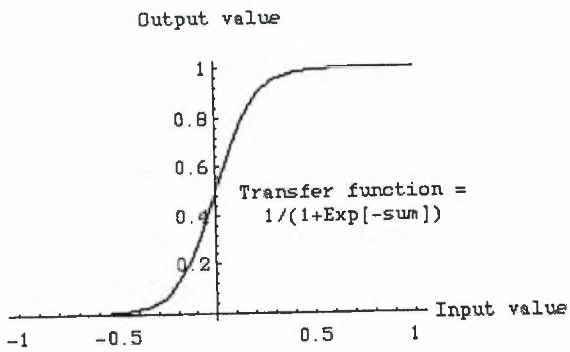


Figure 2.2 Sigmoid Transfer Function.

Figure 2.2 represents sigmoid curve. That curve approaches a minimum and maximum value at the asymptotes. It is common for this curve to be called a sigmoid when it ranges between 0 and 1, and a hyperbolic tangent when it ranges between -1 and 1. Mathematically, the exciting feature of these curves is that both the function and its derivatives are continuous. This option works fairly well and is often the transfer function of choice.

Prior to applying the transfer function, uniformly distributed random noise may be added. The source and amount of this noise is determined by the learning mode of a given network paradigm.

2.3.1.4. Scaling and Limiting

After the processing element's transfer function, the result can pass through additional processes which scale and limit. This scaling simply multiplies a scale factor times the transfer value, and then adds an offset. Limiting is the mechanism which insures that the scaled result does not exceed an upper or lower bound. This limiting is in addition to the hard limits that the original transfer function may have performed. This type of scaling and limiting is mainly used in topologies to test biological neuron models.

2.3.1.5. Output Function (Competition)

Each processing element is allowed one output signal which it may output to hundreds of other neurons. This is just like the biological neuron, where there are many inputs and only one output action. Normally, the output is directly equivalent to the transfer function's result. Some network topologies, however, modify the transfer result to incorporate competition among neighbouring processing elements. Neurons are allowed to compete with each other, inhibiting processing elements unless they have great strength. Competition can occur at one or both of two levels. First, competition determines which artificial neuron will be active, or provides an output. Second, competitive inputs help determine which processing element will participate in the learning or adaptation process.

2.3.1.6. Error Function and Back-Propagated Value

In most learning networks the difference between the current output and the desired output is calculated. This raw error is then transformed by the error function to match particular network architecture. The most basic architectures use this error directly, but some square the error while retaining its sign, some cube the error, and other paradigms modify the raw error to fit their specific purposes. The artificial neuron's error is then typically propagated into the learning function of another processing element. This error term is sometimes called the current error. The current error is typically propagated backwards to a previous layer. Yet, this back-propagated value can be either the current error, the current error scaled in some manner (often by the derivative of the transfer function), or some other desired output depending on the network type. Normally, this back-propagated value, after being scaled by the learning function, is multiplied against each of the incoming connection weights to modify them before the next learning cycle.

2.3.1.7. Learning Function

The purpose of the learning function is to modify the variable connection weights on the inputs of each processing element according to some neural based algorithm. This process of changing the weights of the input connections to achieve some desired result can also be called the adoption function, as well as the learning mode. There are two types of learning: supervised and unsupervised. Supervised learning requires a teacher. The teacher may be a training set of data or an observer who grades the performance of the network results. Either way, having a teacher is learning by reinforcement. When there is no external teacher, the system must organize itself by some internal criteria designed into the network. This is learning by doing.

2.3.2 Electronic Implementation of Artificial Neurons

In currently available software packages these artificial neurons are called "processing elements" and have many more capabilities than the simple artificial neuron described above. . Figure 2.3 is a more detailed schematic of this still simplistic artificial neuron.

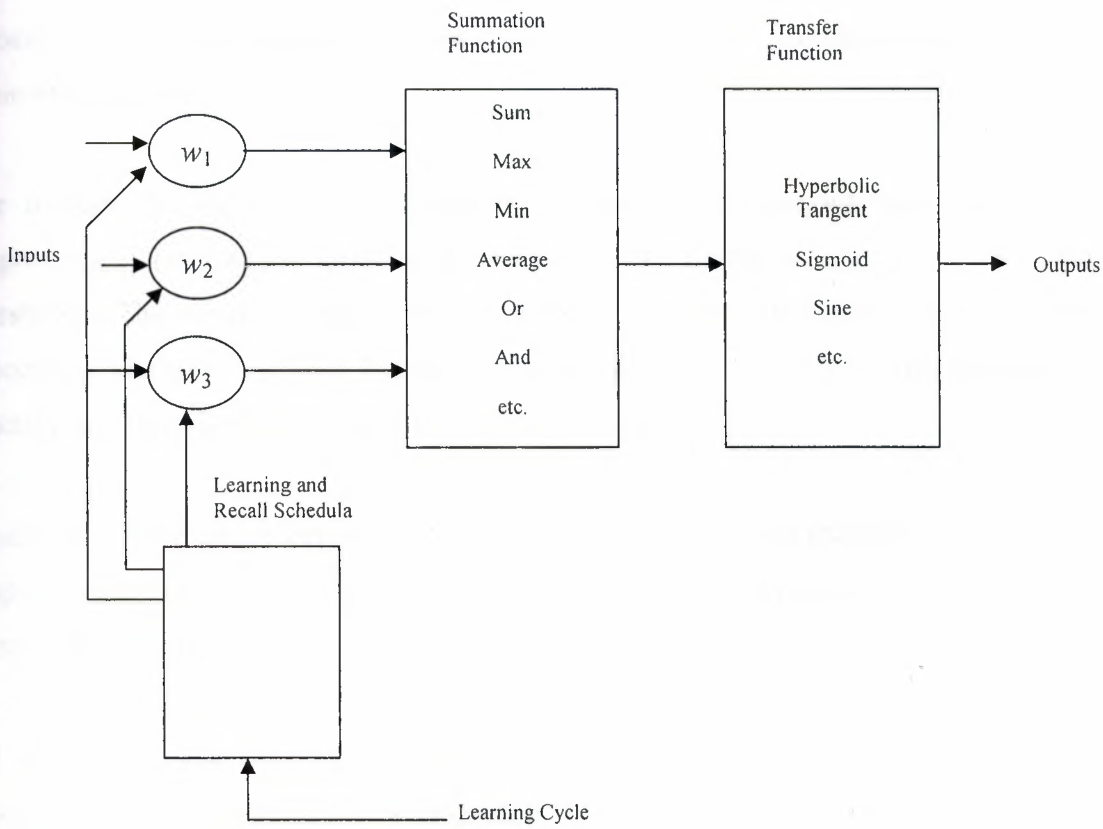


Figure 2.3 A Model of a "Processing Element"

Inputs enter into the processing element from the upper left. The first step is for each of these inputs to be multiplied by their respective weighting factor (w_n). Then these modified inputs are fed into the summing function, which usually just sums these products. Yet, many different types of operations can be selected. These operations could produce a number of different values which are then propagated forward; values such as the average, the largest, the smallest, the ORed values, the ANDed values, etc. Furthermore, most commercial development products allow software engineers to create their own summing functions via routines coded in a higher level language (C is commonly supported). Sometimes the summing function is further complicated by the addition of an activation function which enables the summing function to operate in a time sensitive way.

Either way, the output of the summing function is then sent into a transfer function. This function then turns this number into a real output via some algorithm. It is this algorithm that takes the input and turns it into a zero or a one, a minus one or a one, or some other number.

The transfer functions that are commonly supported are sigmoid, sine, hyperbolic tangent, etc. This transfer function also can scale the output or control its value via thresholds. The result of the transfer function is usually the direct output of the processing element. Sigmoid transfer function takes the value from the summation function and turns it into a value between zero and one.

Finally, the processing element is ready to output the result of its transfer function. This output is then input into other processing elements, or to an outside connection, as dictated by the structure of the network.

All artificial neural networks are constructed from this basic building block - the processing element or the artificial neuron. It is variety and the fundamental differences in these building blocks which partially cause the implementing of neural networks to be an "art."

2.4 Neural Network Learning

The brain basically learns from experience. Neural networks are sometimes called machine-learning algorithms, because changing of its connection weights (training) causes the network to learn the solution to a problem [4]. The strength of connection between the neurons is stored as a weight-value for the specific connection. The system learns new knowledge by adjusting these connection weights. The learning ability of a neural network is determined by its architecture and by the algorithmic method chosen for training.

2.4.1 Definition of Learning

In as much as a great variety of human experience can be described as learning, the term machine learning is sometimes obscure. A somewhat more focused definition suggested by Herbert Simon (1983) is based on the notion of change:

> Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently and more effectively the next time [5].

Learning can refer to either acquiring new knowledge or enhancing or fining skills. Learning new knowledge includes acquisition of significant concepts, understanding of their meanings and relationships to each other and the domain concerned. The new knowledge should be assimilated and put a mentally usable form before it can be called "learned." Thus, knowledge acquisition is defined as learning new symbolic information combined with the ability to use that information effectively.

2.4.2 Classifications of Neural Network Learning

Once a network has been structured for a particular application, that network is ready to be trained. To start this process the initial weights are chosen randomly. Then, the training, or learning, begins.

There are two approaches to learning - supervised and unsupervised. Supervised learning involves a mechanism of providing the network with the desired output either by manually "grading" the network's performance or by providing the desired outputs with the inputs. Unsupervised learning is where the network has to make sense of the inputs without outside help.

2.4.2.1 Supervised Learning

Supervised learning algorithms utilize the information on the class membership of each training instance. This information allows supervised learning algorithms to detect pattern misclassifications as a feedback to themselves. Error information contributes to the learning process by rewarding accurate classifications and/or punishing misclassifications-a process known as credit and blame assignment. It also helps eliminate implausible hypothesis [3]. In supervised learning, the network updates itself by repeatedly comparing a given correct input until it gets the feature of that input. Like: Perception, Back propagation, Hopfield, etc.

The vast majority of artificial neural network solutions have been trained with supervision. In this mode, the actual output of a neural network is compared to the desired output. Weights, which are usually randomly set to begin with, are then adjusted by the network so that the next iteration, or cycle, will produce a closer match between the desired and the actual output. The learning method tries to minimize the current errors of all processing elements. This global error reduction is created over time by continuously modifying the input weights until acceptable network accuracy is reached. With supervised learning, the artificial neural network must be trained before it becomes useful. Training consists of presenting input and output data to the network. This data is often referred to as the training set. That is, for each input set provided to the system, the corresponding desired output set is provided as well. In most applications, actual data must be used. This training phase can consume a lot of time. In prototype systems, with inadequate processing power, learning can take weeks. This training is considered complete when the neural network reaches an user defined performance level. This level signifies that the network has achieved the desired statistical accuracy as it produces the required outputs for a given sequence of inputs.

When no further learning is necessary, the weights are typically frozen for the application. Some network types allow continual training, at a much slower rate, while in operation. This helps a network to adapt to gradually changing conditions [4].

After a supervised network performs well on the training data, then it is important to see what it can do with data it has not seen before. If a system does not give reasonable outputs for this test set, the training period is not over. Indeed, this testing is critical to insure that the network has not simply memorized a given set of data but has learned the general patterns involved within an application.

2.4.2.2 Unsupervised Learning

Unsupervised learning algorithms use unlabeled instances. They blindly or heuristically process them. Unsupervised learning algorithms often have less computational complexity and less accuracy than supervised learning algorithms. Unsupervised learning algorithms can be designed to learn rapidly. This makes unsupervised learning practical in many high-speed, real-time environments, where we may not have enough time and information to apply supervised techniques. Unsupervised learning has also been used for scientific discovery. In this application, the learner should focus its attention on interesting concepts, and the value of interestingness is determined in a heuristic manner [3]. In unsupervised learning, the network learns by "rules" rather than by inputs. Like: Kohenen's, Competitive learning, ART, etc.

Unsupervised learning is the great promise of the future. It shouts that computers could someday learn on their own in a true robotic sense [4]. This promising field of unsupervised learning is sometimes called self-supervised learning. These networks use no external influences to adjust their weights. Instead, they internally monitor their performance. These networks look for regularities or trends in the input signals, and makes adaptations according to the function of the network. Even without being told whether it's right or wrong, the network still must have some information about how to organize itself. This information is built into the network topology and learning rules.
An unsupervised learning algorithm might emphasize cooperation among clusters of processing elements. In such a scheme, the clusters would work together. If some external input activated any node in the cluster, the cluster's activity as a whole could be

increased. Competition between processing elements could also form a basis for learning. Training of competitive clusters could amplify the responses of specific groups to specific stimuli. As such, it would associate those groups with each other and with a specific appropriate response. Normally, when competition for learning is in effect, only the weights belonging to the winning processing element will be updated.
At the present state of the art, unsupervised learning is not well understood and is still the subject of research. This research is currently of interest to the government because military situations often do not have a data set available to train a network until a conflict arises.

2.4.3 Learning Rates

The rate at which ANNs learn depends upon several controllable factors. In selecting the approach there are many trade-offs to consider. Obviously, a slower rate means a lot more time is spent in accomplishing the off-line learning to produce an adequately trained system. With the faster learning rates, however, the network may not be able to make the fine discriminations possible with a system that learns more slowly. Researchers are working on producing the best of both worlds.

Generally, several factors besides time have to be considered when discussing the off-line training task, which is often described as "tiresome." Network complexity, size, paradigm selection, architecture, type of learning rule or rules employed, and desired accuracy must all be considered. These factors play a significant role in determining how long it will take to train a network. Changing any one of these factors may either extend the training time to an unreasonable length or even result in an unacceptable accuracy.

Most learning functions have some provision for a learning rate, or learning constant. Usually this term is positive and between zero and one. If the learning rate is greater than one, it is easy for the learning algorithm to overshoot in correcting the weights, and the network will oscillate. Small values of the learning rate will not correct the current error as quickly, but if small steps are taken in correcting errors, there is a good chance of arriving at the best minimum convergence [4].

2.4.4 Learning Laws

Many learning laws are in common use. Most of these laws are some sort of variation of the best known and oldest learning law, Hebb's Rule [4]. Research into different learning functions continues as new ideas routinely show up in trade publications. Some researchers have the modeling of biological learning as their main objective. Others are experimenting with adaptations of their perceptions of how nature handles learning. Either way, man's understanding of how neural processing actually works is very limited. Learning is certainly more complex than the simplifications represented by the learning laws currently developed. A few of the major laws are presented as examples.

- **Hebb's Rule:** The first, and undoubtedly the best known, learning rule was introduced by Donald Hebb. The description appeared in his book *The Organization of Behavior* in 1949. His basic rule is: If a neuron receives an input from another neuron and if both are highly active (mathematically have the same sign), the weight between the neurons should be strengthened.

- **Hopfield Law:** It is similar to Hebb's rule with the exception that it specifies the magnitude of the strengthening or weakening. It states, "if the desired output and the input are both active or both inactive, increment the connection weight by the learning rate, otherwise decrement the weight by the learning rate."

- **The Delta Rule:** This rule is a further variation of Hebb's Rule. It is one of the most commonly used. This rule is based on the simple idea of continuously modifying the strengths of the input connections to reduce the difference (the delta) between the desired output value and the actual output of a processing element. This rule changes the synaptic weights in the way that minimizes the mean squared error of the network. This rule is also referred to as the Widrow-Hoff Learning Rule and the Least Mean Square (LMS) Learning Rule.

  The way that the Delta Rule works is that the delta error in the output layer is transformed by the derivative of the transfer function and is then used in the previous neural layer to adjust input connection weights. In other words, this error is back-propagated into previous layers one layer at a time. The process of backpropagating the network errors continues until the first layer is reached. The network type called Feedforward, Back-propagation derives its name from this method of computing the error term.

When using the delta rule, it is important to ensure that the input data set is well randomized. Well ordered or structured presentation of the training set can lead to a network which can not converge to the desired accuracy. If that happens, then the network is incapable of learning the problem.

- **The Gradient Descent Rule**: This rule is similar to the Delta Rule in that the derivative of the transfer function is still used to modify the delta error before it is applied to the connection weights. Here, however, an additional proportional constant tied to the learning rate is appended to the final modifying factor acting upon the weight. This rule is commonly used, even though it converges to a point of stability very slowly. It has been shown that different learning rates for different layers of a network help the learning process converge faster. In these tests, the learning rates for those layers close to the output were set lower than those layers near the input. This is especially important for applications where the input data is not derived from a strong underlying model.

- **Kohonen's Learning Law:** This procedure, developed by Teuvo Kohonen, was inspired by learning in biological systems. In this procedure, the processing elements compete for the opportunity to learn, or update their weights. The processing element with the largest output is declared the winner and has the capability of inhibiting its competitors as well as exciting its neighbours. Only the winner is permitted an output, and only the winner plus its neighbours are allowed to adjust their connection weights.

Further, the size of the neighbourhoods can vary during the training period. The usual paradigm is to start with a larger definition of the neighbourhoods, and narrow in as the training process proceeds. Because the winning element is defined as the one that has the closest match to the input pattern, Kohonen networks model the distribution of the inputs. This is good for statistical or topological modelling of the data and is sometimes referred to as self-organizing maps or self-organizing topologies.

2.5 Back Propagation

The feed forward, back-propagation architecture was developed in the early 1970s by several independent sources (Werbor; Parker; Rumelhart, Hinton and Williams). This independent co-development was the result of a proliferation of articles and talks at various conferences that stimulated the entire industry. Currently, this synergistically developed back-propagation architecture is the most popular, effective, and easy to learn model for complex, multi-layered networks. This network is used more than all other combined. It is used in many different types of applications. This architecture has spawned a large class of network types with many different topologies and training methods. Its greatest strength is in non-linear solutions to ill-defined problems [3].

The back propagation network is probably the most well known and widely used among the current types of neural network systems available. In contrast to earlier work on perceptron, the back propagation network is a multilayer feed forward network with a different transfer function in the artificial neuron and a more powerful learning rule.

The learning rule is known as back propagation, which is a kind of gradient descent technique with backward error (gradient) propagation, as depicted in Figure 2.4. The training instance set for the network must be presented many times in order for the interconnection weights between the neurons to settle into a state for correct classification of input patterns. While the network can recognize patterns similar to those they have learned, they do not have the ability to recognize new patterns. This is true for all supervised learning networks. In order to recognize new patterns, the network needs to be retrained with these patterns along with previously known patterns. If only new patterns are provided for retraining, then old patterns may be forgotten. In this way, learning is not incremental over time. This is a major limitation for supervised learning networks. Another limitation is that the back propagation network is prone to local minima, i.e., the error becomes smaller then larger then smaller and so forth, at one location, just like any other gradient descent algorithm, also the training time is long [3].

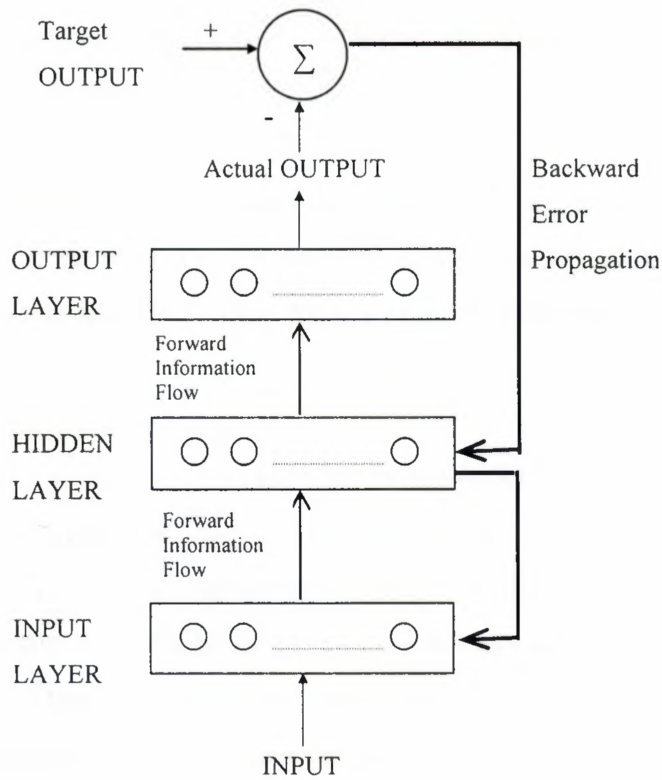


Figure 2.4 The backpropagation network

The typical back propagation network has an input layer, an output layer, and at least one hidden layer. There is no theoretical limit on the number of hidden layers but typically there is just one or two. The in and out layers indicate the flow of information during recall. Recall is the process of putting input data into a trained network and receiving the answer. Back propagation is not used during recall, but only when the network is learning a training set [4]. The number of layers and the number of processing element per layer are important decisions. These parameters to a feed forward, back propagation topology are also the most ethereal. They are the art of the network designer. There is no quantifiable, best answer to the layout of the network for any particular application. There are only general rules picked up over time and followed by most researchers and engineers applying this architecture of their problems.

- Rule One: As the complexity in the relationship between the input data and the desired output increases, then the number of the processing elements in the hidden layer should also increase.

- Rule Two: If the process being modeled is separable into multiple stages, then additional hidden layer(s) may be required. If the process is not separable into stages, then additional layers may simply enable memorization and not a true general solution.

- Rule Three: The amount of training data available sets an upper bound for the number of processing elements in the hidden layers. To calculate this upper bound, use the number of input output pair examples in the training set and divide that number by the total number of input and output processing elements in the network. Then divide that result again by a scaling factor between five and ten. Larger scaling factors are used for relatively noisy data. Extremely noisy data may require a factor of twenty or even fifty, while very clean input data with an exact relationship to the output might drop the factor to around two. It is important that the hidden layers have few processing elements. Too many artificial neurons and the training set will be memorized. If that happens then no generalization of the data trends will occur, making the network useless on new data sets.

Once the above rules have been used to create a network, the process of teaching begins. This teaching process for a feed forward network normally uses some variant of the Delta Rule, which starts with the calculated difference between the actual outputs and the desired outputs. Using this error, connection weights are increased in proportion to the error times a scaling factor for global accuracy. Doing this for an individual node means that the inputs, the output, and the desired output all have to be present at the same processing element. The complex part of this learning mechanism is for the system to determine which input contributed the most to an incorrect output and how does that element get changed to correct the error. An inactive node would not contribute to the error and would have no need to change its weights. To solve this problem, training inputs are applied to the input layer of the network, and desired outputs are compared at the output layer. During the learning process, a forward sweep is made through the network, and the output of each element is computed layer by layer. The difference between the output of the final layer and the desired output is back-propagated to the previous layer(s), usually modified by the derivative of the transfer function, and the connection weights are normally adjusted using the Delta Rule. This process proceeds for the previous layer(s) until the input layer is reached.

There are many variations to the learning rules for back-propagation network. Different error functions, transfer functions, and even the modifying method of the derivative of the transfer function can be used. The concept of momentum error was introduced to allow for more prompt learning while minimizing unstable behavior. Here, the error function, or delta weight equation, is modified so that a portion of the previous delta weight is fed through to the current delta weight. This acts, in engineering terms, as a low-pass filter on the delta weight terms since general trends are reinforced whereas oscillatory behaviour is cancelled out. This allows a low, normally slower, learning coefficient to be used, but creates faster learning.

Another technique that has an effect on convergence speed is to only update the weights after many pairs of inputs and their desired outputs are presented to the network, rather than after every presentation. This is referred to as cumulative back-propagation because the delta weights are not accumulated until the complete set of pairs is presented. The number of input-output pairs that are presented during the accumulation is referred to as an epoch. This epoch may correspond either to the complete set of training pairs or to a subset [4].

2.5.1 The Back Propagation Algorithm

The back propagation network consists of one input layer, one output layer, and one or more hidden layers. If n bits or n values describe the input pattern, then there should be n input units to accommodate it. The number of output units, is likewise determined by how many bits or values are involved in the output pattern. Theoretical guidance exists [3] for determining the numbers of hidden layers and hidden units. They can be recruited or pruned as indicated by the network performance. Typically, the network is fully connected between and only between adjacent layers as shown Figure 2.5. The back propagation algorithm (Rumelhart, Hinton, and Williams 1986) is formulated below.

This is the simple three layer back propagation model. Each neuron is represented by a circle and each interconnection, with its associated weight, by arrow. The neurons labelled *b* are biased neurons. Normalization of the input data prior to training is necessary. The values of the input data into the input layer must be in the range (0-1). The stages of the feed forward calculations can be described according to the layers. The suffixes *i, h, j* are used for input, hidden and output respectively.

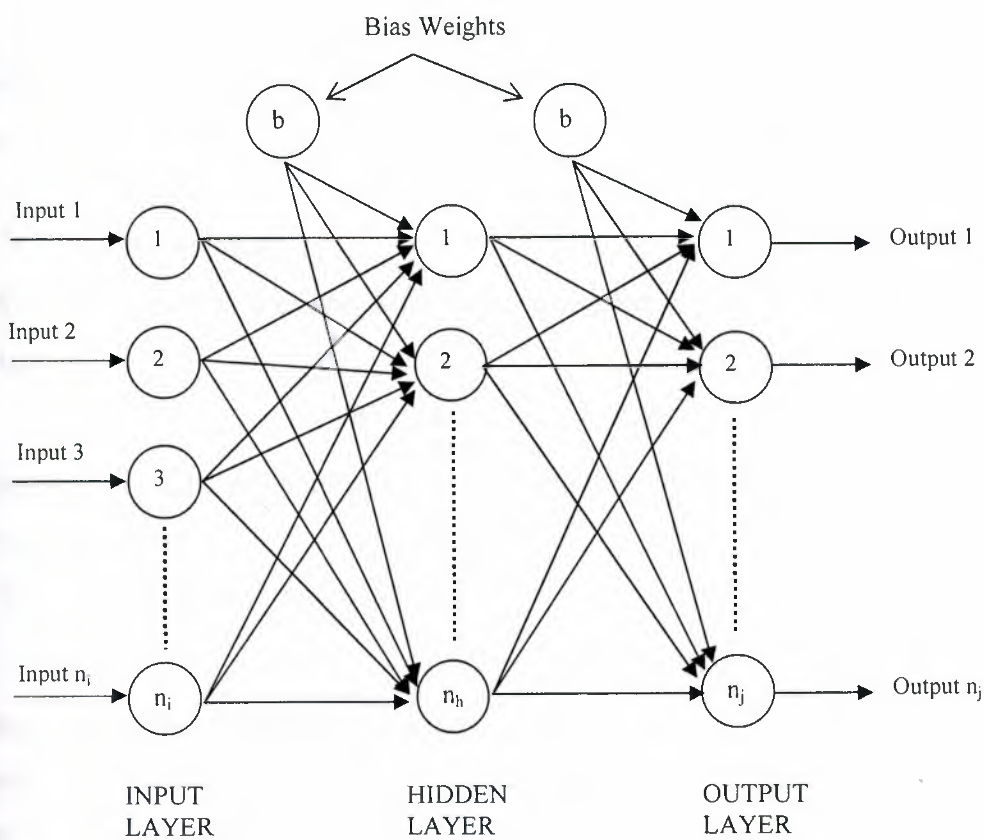


Figure 2.5 Back Propagation Network Structure

$n_i \longrightarrow$ number of input layer nodes

$n_h \longrightarrow$ number of hidden layer nodes

$n_j \longrightarrow$ number of output layer nodes

▪ Weight Initialization

Set all weights and node thresholds to small random numbers. Note that the node threshold is the negative of the weight from the bias unit (whose activation level is fixed at 1).

▪ Calculation of Activation

1. The activation level of an input unit is determined by the instance presented to the network.

2. The activation level Oj of a hidden and O_k of an output unit can be determined by

$$O_j = F\left(\sum W_{ji} O_i - \theta_j\right) \tag{2.2}$$

$$O_k = F\left(\sum W_{kj} O_j - \theta_k\right) \tag{2.2a}$$

where W_{ji} is the weight from an input O_i, θ_j is the node threshold, and F is a sigmoid function:

▪ Weight Training

1. Start at the output units and work backward to the hidden layers recursively. Adjust weights by

$$W_{ji}(t+1) = W_{ji}(t) + \Delta W_{ji} \tag{2.3}$$

where $W_{ji}(t)$ is the weight from unit i to unit j at time t (or the iteration) and $\Delta\ W_{ji}$. is the weight adjustment.

2. The weight change is computed by

$$\Delta W_{ji} = \eta \delta_j O_i \tag{2.4}$$

where η is a trial-independent learning rate ($0 < \eta < 1$, e.g., 0.3) and δ_j is the error gradient at unit j. Convergence is sometimes faster by adding a momentum term (α), also to avoid local minima:

$$W_{ji}(t+1) = W_{ji}(t) + \eta \delta_j O_i + \alpha\left[W_{ji}(t) - W_{ji}(t-1)\right] \tag{2.5}$$

where $0 < \alpha < 1$.

3. The error gradient is given by:

- For the output units:

$$\delta_j = O_j(1 - O_j)(T_j - O_j) \tag{2.6}$$

3. PREDICTION OF PRODUCT QUALITY USING NEURAL NETWORKS

3.1 Overview

In this chapter, development of neural network system of product quality prediction is described. A structure of neural network system to predict product quality will be presented. Selection of process variables that have influence to product quality is determined. The main steps for development of neural network system to predict naphtha cut point will be explained in details.

3.2 Analysis of Technological Process

Petroleum industry is one of the most prolific and dynamic industries of the modern civilization. Because of a highly competitive market and stringent environmental laws, strict quality control of refinery products is a must. Crude distillation unit (CDU) is one through which entire crude entering a refinery must be processed. So a close monitoring and control of CDU product properties will help us in controlling the properties of final refinery products.

A typical distillation column processes about hundred tons of materials per hour, complying with severe specifications on the purity of the distillate, irrespective of the quality of the raw product. Because of the very long time constants involved in the operation of such processes, the early fault detection is an important problem: a malfunction which is not detected very shortly after its inception may result in wasting tens of hours of operation. Fault detection means prediction of product quality. The difficulty of prediction of product quality arises from the fact that the number of measured variables is usually small as compared to the number of state variables, that they are noisy, and that trends observed in the measurements are often ambiguous.

The yield and properties or qualities of the fractions are determined by the Final Boiling Point or Cut Point settings used to operate the crude distillation unit (CDU). The relationships between the set of Cut Points and properties of the resulting fractions are extremely non-linear [6]. The existence of highly nonlinear relationships between the process variables (inputs) and the product stream properties (outputs) have been stuck all efforts to come up with reliable mathematical models for large scale CDU of an oil refinery. In order to solve this problem in a timely fashion, a continuous on-line method for predicting product stream properties and pertinence to column operation of the oil refinery are needed. The proposed neural network architectures using back propagation algorithm can accurately predict various properties associated with crude oil production.

3.3 Structure of Neural Networks System for the Prediction of Naphtha Cut Points

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 [6]. Some very useful predictions can be constructed in processing plants using algorithms whose coefficients are discovered through training. Figure 3.1 is a graphical representation of the artificial neural network structure. A neural network predictor is built by discovering the weights. W_1K_1 through W_1K_n are the corresponding weights of the first neuron. The output Q_p is the predicted inferred process stream property (95%, 90% cut-point, etc.).

The coefficients of the model are discovered by training a neural network program using back propagation algorithms. The input of the neural network consist of plant data such as density (API), temperature and flow rate 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 artificial neural network coefficients is training the network [8].

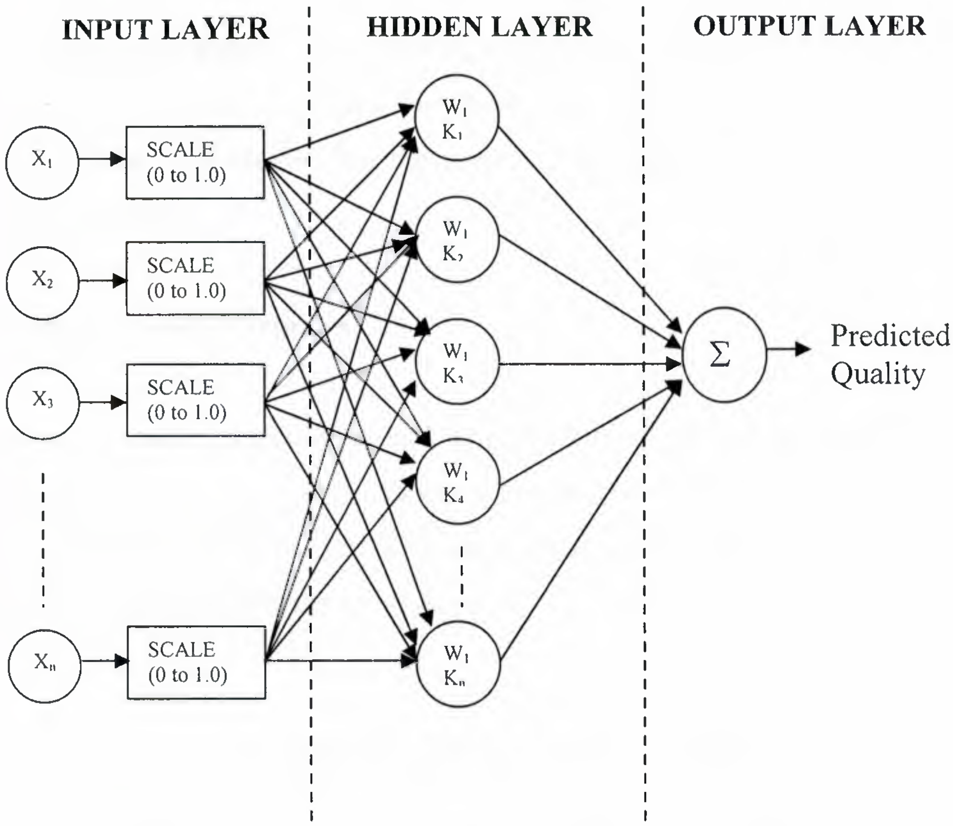


Figure 3.1 Graphical representation of ANN structure

3.3.1 Defining Training Data Set

Neural networks will not be an accurate predictor if the operating input/output data are outside of 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.

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 the process or as the result of inaccuracies in data acquisition techniques. A large data set will average out various inaccuracies within a system.

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 the interest to complete the gaps within the required data.

3.3.2 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.

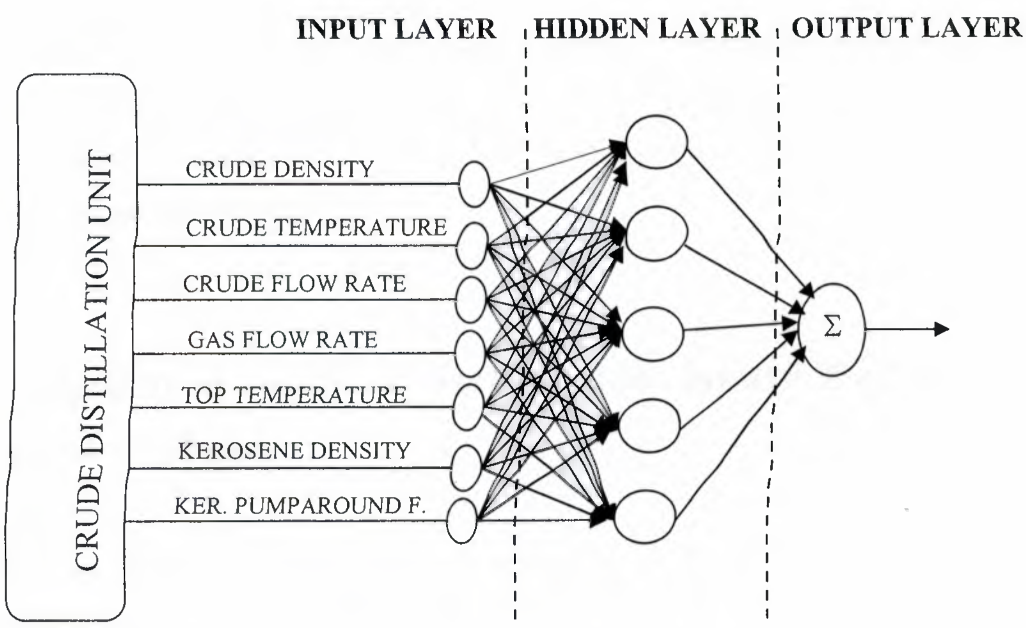


Figure 3.2 Process Variable Selection in Crude Distillation Unit

Figure 3.2 shows the selected process variables to predict 95% Naphtha Cut-point in crude distillation unit. Density of crude oil and kerosene, flow rates of crude oil, gas and kerosene pump around, top temperature and temperature of crude oil are chosen as process variables. If those process variables chosen initially do not give the required accuracy of prediction, less important variables should be dropped and other parameters added.

All identified process parameters do not necessarily have an effect on each of the product quality. The important point in selecting process variables 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.

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 large degree of variance. It is possible that the most important parameter that affects a particular product quality keeps the same value in all generated data sets. The neural network program will ignore such a parameter. Thus elimination should not include variables which, from engineering point of view, should have a contribution on the inferred analysis.

3.4 Development of Neural Networks System for the Prediction of Naphtha Cut Points

The major steps that are involved in implementing the ANN predictor are shown in Figure 3.3.

3.4.1 Identifying Application

The first step for construction of a neural network system is the appropriate identification of a potential application. Crude oil as well as all the CDU products is complex mixtures of hydrocarbons, so it is not convenient to characterize them in terms of individual components. Moreover, it is often sufficient to characterize refinery products in terms of certain gross properties such as Cut Point for fractionation of all products, Reid Vapor Pressure for volatile products, Flash Point for light distillate, Pour Point for heavy distillate, etc [9].

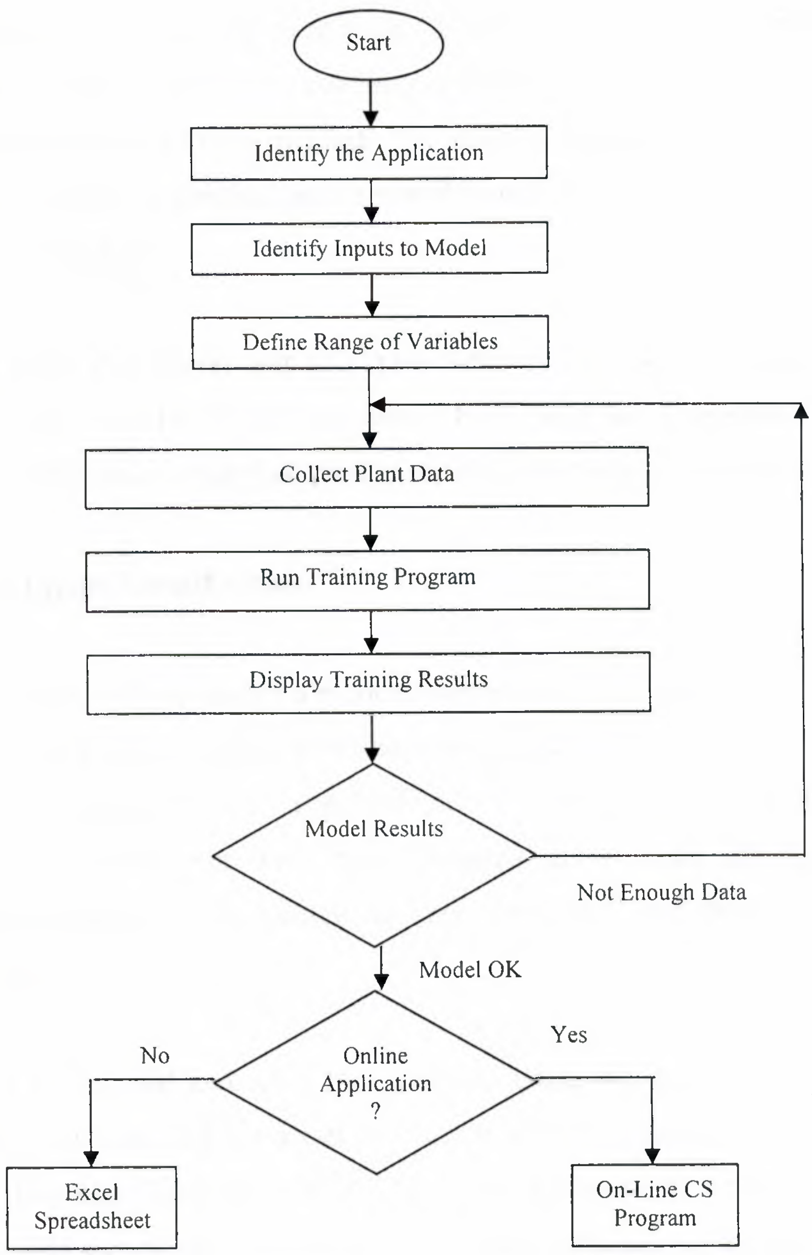


Figure 3.3 Major Steps for Implementing of ANN in the CDU Process

So a close monitoring and control of CDU product properties will help in controlling the properties of final refinery products. The qualities and properties of the fractions are determined by Cut Point settings used to operate CDU. The relationships between the set of Cut Points and properties of the resulting fractions are extremely non-linear. The existence of highly nonlinear relationships between the process variables (inputs) and the product stream properties (outputs) have been stuck all efforts to come up with reliable mathematical models for large scale CDU of an oil refinery. In order to solve this problem in a timely fashion, a continuous on-line method for predicting product stream properties of the CDU is needed. The proposed neural network architectures using back propagation algorithm can accurately predict various properties associated with crude oil production.

In this thesis crude distillation unit of Tüpraş Refinery, in İzmit, Turkey is chosen as target to predict Naphtha 95% Cut-point. Once Naphtha Cut-point is predicted successfully, model could be applied prediction of other product stream properties.

3.4.2 Model Inputs Identification

The neural network will not match a random number set. For prediction model to work there must be some relationships between input/output variables [10]. 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 [11].

Density of crude oil and kerosene, flow rates of crude oil, gas and kerosene pump around, top temperature and temperature of crude oil are chosen as input variables to predict 95% Naphtha Cut-point in CDU of Tüpraş Refinery, in İzmit, Turkey. 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.

3.4.3 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.

3.4.4 Predictor Model Training

For the naphtha 95 % cut point stream properties in CDU of Tüpraş Refinery [12], in İzmit, Turkey, the plant data, including the stream quality desired to predict, are collected in a Microsoft Excel™ spreadsheet for four month period to facilitate data manipulation (Appendix I).

The spreadsheet file is loaded into the Neuroshell™ neural network package and NN-pred in Microsoft Excel™ software package. Both software programs use 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. The change in weight is proportional to that element's effect on the sum-squared error of the network.

Initially, one hidden layer with eight 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 multidimensional surface [14]. By selecting the minimum error to be a very small number (like 10^{-3}) the program will end up in one of the following states:

- Minimum error goal is matched before exceeding the limit on maximum allowed iterations. In this case, the objective of the training is successfully met.

- 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.
- 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.

After training, the test of developed system is carried out. For this reason the statistical data taken from the process is used. The result of training and test are monitoring on-line. In case of satisfactory in obtained test results, the results of modeling of prediction system could be used in on-line application in oil refinery process for predicting naphtha cut points and other product properties

3.5 Summary

The purpose of this chapter was to develop a neural network system for predicting product quality of crude distillation unit, namely naphtha 95 % cut point. To do this, a systematic procedure to construct a neural network model is presented. Selection of appropriate data sets as well as data analysis procedure was discussed. Various steps in the implementation phase of neural network model in the crude oil fractionation process were devoted.

4. MODELLING OF NEURAL NETWORK FOR PREDICTING QUALITY OF NAPHTHA CUT-POINTS

4.1 Overview

In this chapter, the neural network learning structure and the training procedures as well as the results of the modelling for naphtha 95 % cut point will be analyzed.

4.2 Algorithmic Description of Neural Network System for Predicting Naphtha 95 % Cut Point

The increasing quality of technological processes needs the development of efficient control system. The development of such control systems has great importance in the production where great volume of information describing the state of processes to be processed on the finite time interval. This requires developing the system that would process the coming information in parallel and with high level of reliability. The research work shows the systems based on neural networks meet the above requirements.

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 results in an increase in profit.

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 target 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.

In the work, back propagation neural network algorithm is used to predict the product quality of crude distillation unit. The naphtha 95 % cut point property is chosen to predict. The main parameters which have considerable influence to the naphtha cut points are selected as inputs for neural prediction system and tabulated in Table 4.1.

Table 4.1 Selected Process Variables for Predicting Naphtha 95 % Cut Point

Selected Inputs	**Tag**	**Range**
Crude Density (API)	D_c	30.5-37.0
Crude Flow Rate (m^3/day)	F_c	11500-13100
Crude Temperature (°C)	T_c	320-330
Gas Flow Rate (m^3/day)	F_G	7500-10500
Top Temperature (°C)	T_T	135-165
Kerosene Density (API)	D_K	43.5-46.5
Kerosene Pumparound F. (m^3/day)	F_K	1100-1800

Table 4.2 shows the fragment of statistical data taken from CDU of Tüpraş Refinery and characterising relation between selected inputs and naphtha 95% cut point.

Table 4.2 Fragment of Statistical Data of Crude Distillation Unit

DATE	INPUTS							OUTPUT
	T_C	D_C	F_C	F_G	T_T	D_K	F_K	T_K (95%)
01.04.2002	321	35,6	11600	9141	145	45,4	1450	195
02.04.2002	323	35,4	11700	8998	155	45,5	1250	197
03.04.2002	324	35,3	11700	8500	145	45,6	1100	198
04.04.2002	322	35,2	12100	9391	147	45,8	1100	196
05.04.2002	323	35,2	12200	9450	153	45,6	1400	195
06.04.2002	324	34,9	12400	9560	159	44,6	1350	205
07.04.2002	323	34,8	12500	8890	152	44,6	1400	205
08.04.2002	326	34,8	12500	9133	159	44,5	1450	205
09.04.2002	325	36,2	12100	9513	163	44,5	1750	205
10.04.2002	324	36,6	12000	10359	157	44,9	1700	204
11.04.2002	323	36,6	12100	10271	149	45,1	1500	203
12.04.2002	325	35,4	12200	9661	148	44,9	1400	208
13.04.2002	324	36,4	11900	9670	144	45,1	1500	205
14.04.2002	325	36,2	11900	9800	157	44,5	1500	205
15.04.2002	326	36,2	11900	9850	152	44,6	1600	204
16.04.2002	322	32,1	12400	8873	143	44	1400	202
17.04.2002	326	32,1	12500	8923	152	44,3	1400	205
18.04.2002	325	32,1	12500	9190	153	44	1350	207
19.04.2002	329	31,3	12500	9181	153	43,9	1350	207
20.04.2002	326	31,1	12400	8876	154	44,4	1500	206

This set in statistical data is the knowledge base for the neural system. Using these input-output data sets (Appendix 1), the learning of neural system is carried out. In Figure 4.1 the neural network learning structure is shown.

Initial values of synaptic weight values in model are generated by random generator with uniform distribution within the interval from 0 to 1. These weights are used to generate three layer neural networks. The values of selected input parameters of the technological process are entered to the neural network input. These signals are processed by the network and its output compared with the target signals describing naphtha cut points and deviation is calculated. The value of deviation is used by learning algorithm for correction weight parameters of neural networks.

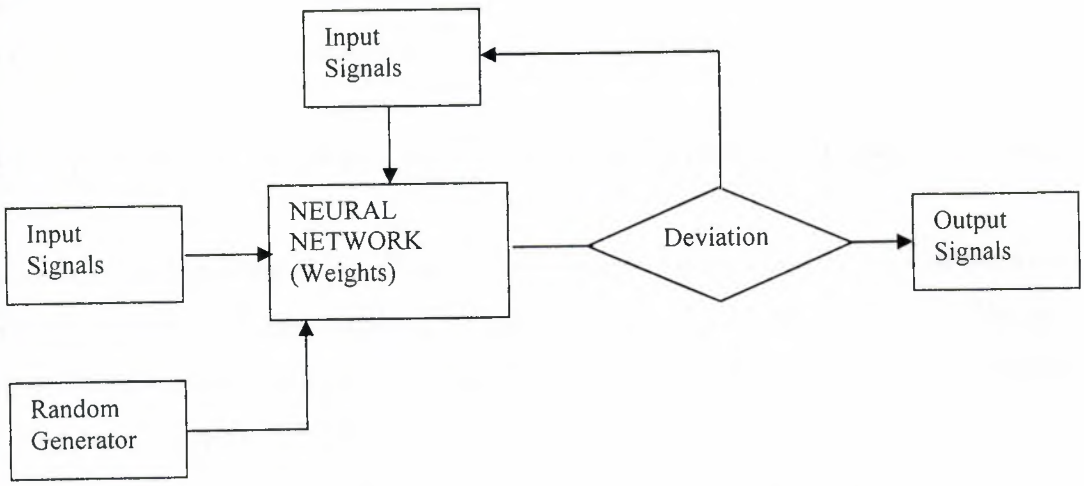


Figure 4.1 The Neural Network Learning Structure

In the model the back propagation algorithm is used for learning parameters of neural system. This process is repeated for all input data. The learning is continued until the value of deviation will be acceptable small for all input-output training pairs.

The trained parameter values are used in controller for predicting naphtha cut point quality.

4.3 Analysis of Obtained Results

Modelling 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, three-layer networks consisting of a input layer, a single hidden layer and an output layer have proved to be adequate.

To demonstrate the modelling capability of a back propagation network, 120 data sets were analyzed. Each data set consisted of seven process variables as inputs to the model and one product quality (Naphtha 95% cut point) as an output. A total of 90 data sets were used in training phase and 30 data sets were used in verification phase.

While developing neural prediction system, two different software package were used; the Neuroshell software package and the NN-pred, a Microsoft Excel™ software package. Modelling by these two packages will be presented below.

4.3.1 Prediction of Naphtha 95 % Cut Point Property Using Neuroshell

Figure 4.2 illustrates the menu of neuroshell system. First, characteristics of the system defined as seven process variables as input to model and one product quality as an output. Then each characteristic is loaded to package as sample cases. Out of 120 data sets, 90 data sets were utilized for training.

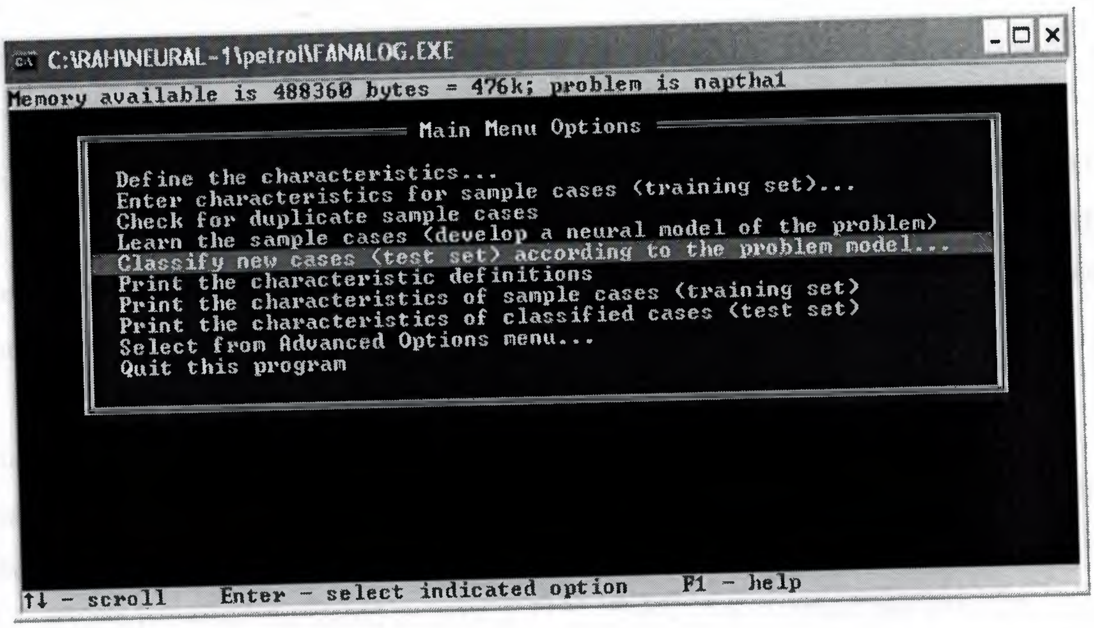


Figure 4.2 The Menu of Neuroshell Software Package

Various number of hidden neuron in a single hidden layer has been tested. For each different number of hidden neuron, learning option was chosen from main menu and was trained. Figure 4.3 illustrates fragment of learning process in neuroshell package.

Learning events completed when system achieved maximum error goal for testing phase. If maximum error obtained is not acceptable, learning process will be restarted using the random weight of previous mode.

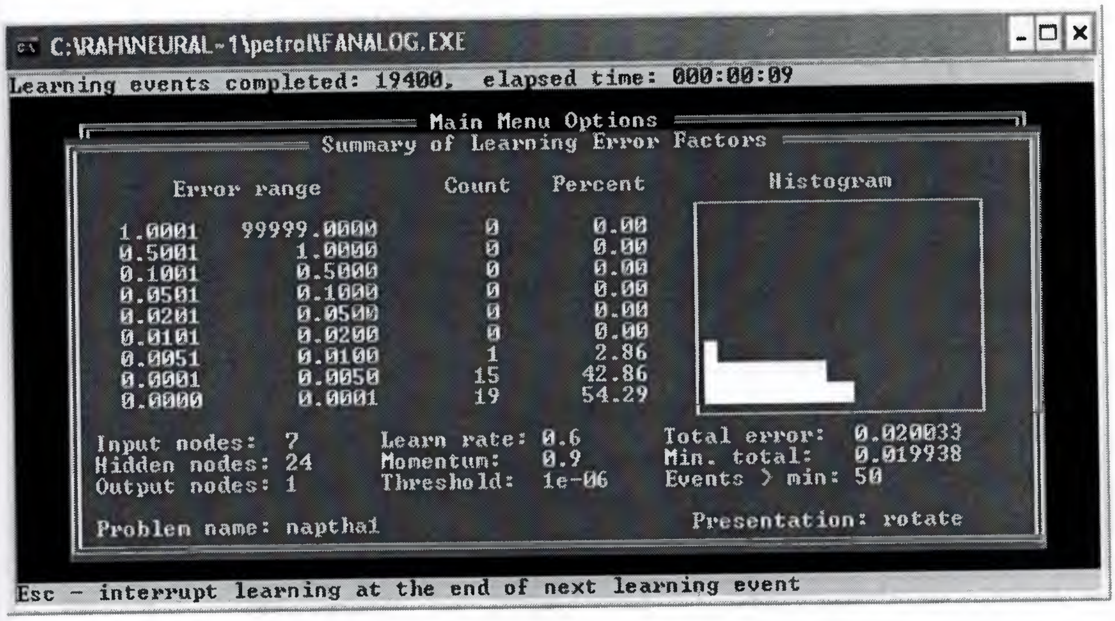


Figure 4.3 A Fragment of Learning Process in Neuroshell Software Package

The most significant results obtained in training modes with eight, ten, twelve and twenty hidden neurons. In all variations, learning rate and momentum values were kept constant as 0.6 and 0.9 respectively. Results of verification phase for 30 data sets are shown in Table 4.3.

Starting with a single hidden layer consisting of eight neurons as a first case, number of hidden neuron was increased to achieve the desired error goal of 0.01. A typical value used for laboratory repeatability for the naphtha 95 % cut point is 1.7 °C. This should be the maximum error for good accuracy [12]. Table 4.4 summarizes the simulation results.

Model with eight hidden neuron achieved very good result in terms of duration of the learning event that is completed within only 90s, the shortest time. But it could not achieve the good result in terms of maximum error value that was 2.52 °C. Although the learning event consumed more time by increasing the number of hidden neurons, maximum error value of training modes decreased. This does not indicate that increasing the number of hidden neurons causes decreasing maximum error value and increasing duration of learning process.

Table 4.3 Results Obtained in Verification Phase Using Neuroshell

Actual Value(°C)	20 Neuron		12 Neuron		10 Neuron		8 Neuron	
	Predicted Value (°C)	MSE	Predicted Value (°C)	MSE	Predicted Value (°C)	MSE	Predicted Value (°C)	MSE
205	204,94	0,06	205,03	0,03	205,27	0,26	204,8	0,20
205	204,93	0,07	204,98	0,02	205,05	0,05	205,03	0,03
204	203,73	0,26	203,99	0,01	204,07	0,07	203,95	0,05
205	204,76	0,23	204,84	0,16	205,08	0,08	204,97	0,03
204	203,89	0,11	204	0,00	204,07	0,07	204,02	0,02
205	205,03	0,03	204,95	0,05	205,05	0,05	205,17	0,17
207	206,96	0,04	206,99	0,01	207,07	0,07	207,02	0,02
206	205,92	0,08	206,06	0,06	206,08	0,08	205,9	0,10
208	207,97	0,03	208,07	0,07	208,03	0,03	208,1	0,10
202	201,94	0,06	202,03	0,03	202,06	0,06	202,1	0,10
205	204,91	0,09	204,85	0,15	205,11	0,11	205,03	0,03
200	199,9	0,10	199,95	0,05	200,13	0,13	199,96	0,04
202	201,91	0,09	201,91	0,09	202,05	0,05	202,02	0,02
202	201,91	0,09	202,01	0,01	202,06	0,06	201,97	0,03
203	202,95	0,05	203,02	0,02	203,06	0,06	203,01	0,01
201	200,88	0,12	201	0,00	201,06	0,06	201,1	0,10
196	195,84	0,16	195,87	0,13	196,07	0,07	196,07	0,07
194	193,69	0,32	193,83	0,18	194,07	0,07	194,05	0,05
196	195,83	0,17	195,82	0,18	196,09	0,09	196,06	0,06
196	195,83	0,17	195,7	0,31	196,1	0,10	196,05	0,05
193	193,24	0.24	193,34	0,35	194,24	1,29	195,51	2,62
202	201,52	0,47	201,15	0,84	199,39	2,57	200,04	1,93
208	208,6	0,58	206,83	1,12	208,07	0,07	206,15	1,77
207	206,95	0,05	205,51	1.43	205,39	1,55	204,86	2,06
205	205,59	0,58	206,15	1,12	206,55	1,51	206,55	1,51
199	200,12	1.12	201,1	2,11	200,68	1,70	202,13	3,20
195	194,68	0,33	194,66	0,35	195,07	0,07	194,95	0,05
198	197,66	0,34	197,5	0,50	198,15	0,15	198	0,00
196	195,66	0,35	195,53	0,48	196,15	0,15	195,98	0,02
205	204,78	0,21	204,72	0,27	205,09	0,09	204,96	0,04

Table 4.4 Simulation Results for Naphtha 95 % Cut Point Using Neuroshell

Hidden Neuron			Training Phase	
	Iteration ($x10^3$)	Duration (sec)	Final MSE (%)	Max. Error(°C)
8	505	90	0.62	2.52
10	2,200	430	0.60	2.61
12	2,350	510	0.55	2.10
20	920	360	0.33	1.12

The best model architecture (in terms of better prediction in testing mode) consists of 20 neurons in one hidden layer. When the acceptable error value is 1.7°C for naphtha 95 % cut point, the model gave with twenty neurons 1.12°C error value for training sets. In this model architecture also consumption time for learning process is decreased, compared to the previous architectures.

4.3.2 Prediction of Naphtha 95 % Cut Point Property Using NNinExcell

Again characteristics of the system defined as seven process variables as input to model and one product quality that is naphtha 95 % cut point as an output. Then data sets were loaded in The Data worksheet, 90 data sets were utilized for training and 30 data sets for verifying.

By filling the model parameters in the User Input Page, building model was initialized. A neural network model is basically a set of weights between the layers of the net. At the end of the each run, the final set of weights was saved in the Calc sheet (see Appendix II). The output page of this file showed the values of MSE- Mean Squared error and ARE- Absolute Relative error on the training and validation set (Appendix III).

Starting with a single hidden layer consisting of eight neurons as a first case, number of hidden neuron was increased to achieve the desired error goal of 0.01. The results of modelling were obtained with eight, ten, twelve and twenty hidden neurons in a single hidden layer. In all variations learning rate, momentum and initial weights were kept constant as 0.1, 0.3 and 0.5 respectively. Results of verification phase for 30 data sets are shown in Table 4.5.

The best model architecture (in terms of better prediction in both training and verification modes) consists of ten neurons in one hidden layer. Model results for training and verification phase will be given in Appendix IV. All model architectures were able to achieve an acceptable error in training phase but failed to achieve comparable results in the verification phase except the model that consists of ten neurons where the maximum absolute error was 1.56 °C. 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 4.6 summarizes the simulation results.

Table 4.5 Results Obtained in Verification Phase Using NNinExcell

	8 Neurons	10 Neurons	12 Neurons	20 Neurons
Actual Naphtha 95 %	Predicted Naphtha 95 %	Predicted Naphtha 95 %	Predicted Naphtha 95 %	Predicted Naphtha 95 %
193	195,245	195,254	195,375	195,432
202	202,381	202,266	202,355	202,332
208	206,717	206,740	206,866	206,937
207	206,338	206,391	206,472	206,648
205	205,586	205,588	205,654	205,834
199	200,073	200,088	200,128	200,253
195	196,408	196,387	196,421	196,367
198	200,812	200,750	200,925	200,882
196	197,949	197,970	197,954	197,983
205	204,691	204,689	204,715	204,798
205	206,075	206,077	206,155	206,297
205	206,151	206,124	206,259	206,256
204	204,338	204,296	204,386	204,344
205	205,066	205,028	205,120	205,120
204	204,925	204,890	204,996	205,057
205	205,216	205,214	205,275	205,400
207	206,565	206,593	206,681	206,819
206	205,760	205,758	205,843	205,898
208	205,738	205,741	205,831	205,895
202	202,645	202,648	202,612	202,637
205	202,969	203,013	202,946	203,028
200	202,031	201,993	201,918	201,799
202	201,894	201,902	201,903	201,952
202	203,102	203,058	203,134	203,189
203	204,071	204,054	204,100	204,199
201	199,238	199,225	199,304	199,393
196	196,833	196,869	196,988	197,222
194	194,278	194,351	194,334	194,446
196	195,397	195,464	195,558	195,744
196	196,450	196,466	196,662	196,726

Table 4.6 Simulation Results for Naphtha 95 % Cut Point Using NNinExcel

Hidden Neurons	Training Phase			Verification Phase	
	Epoch	Final MSE (%)	Max. Error(°C)	Final MSE (%)	Max. Error(°C)
8	100	1.050	1.23	1.5631	2.03
	250	0.932	1.21	1.5358	2.13
10	100	1.063	1.27	1.5297	1.56
	250	0.957	1.18	1.5452	1.61
12	100	1.079	1.32	1.5952	1,78
	250	0.948	1.26	1.6135	2.01
20	100	1.145	1.27	1.6031	2.16
	250	0.998	1.24	1.5701	2.34

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 relationship 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 error goal in the training phase should generate a degree of accuracy very close to lab repeatability. A typical value for lab repeatability for the naphtha 95 % cut point is 1.7 °C. 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.

Table 4.7 shows final results for the best models in both software packages Neuroshell and NNinExcel. It is obvious that Neuroshell software package is more improved than NNinExcel in terms of its capability. Although there was a limitation in NNinExcel that was number of hidden neurons cannot exceed more than twenty, the performance of this package was successful. To achieve best model within the short operation time, model parameters that are learning rate and momentum was different in both. While the best model was achieved in Neuroshell by reaching desired error goal, in NNinExcel that was obtained by reaching maximum epoch.

Table 4.7 The Best Models in Neuroshell and NNinExcel

	Neuroshell	**NNinExcel**
No. of Hidden Neurons	20	10
Learning Rate	0.6	0.1
Momentum	0.9	0.3
Iteration	$920x10^3$	100
Max. Error in Testing Phase (^{o}C)	1.12	1.56

4.4 Summary

In this chapter, the training procedures as well as the results of the modelling for naphtha 95 % cut point was analyzed. It was shown that the proposed neural network models predict product quality well within the specified error goals in both training and verification phases.

5. CONCLUSION

In the thesis, back propagation neural network algorithm was used to predict the product quality of crude distillation unit in oil refinery. The naphtha 95 % cut point property is chosen for prediction.

The objective of the proposed work was 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 network to predict those particular product qualities to meet the more stringent market specifications. 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 results in an increase in profit.

Various neural network architectures were proposed for the prediction of product quality of a crude distillation unit. The important parameters involved in acquiring valid data sets were considered. Close attention is paid to the proper selection of the input data. Finally, product quality property, namely, naphtha 95 % cut point was successfully modelled 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 of the different process variables as inputs and product quality as an output. 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 used fall within the range that was used to create the model. If parameters fall out of range, then the product quality value is questionable and it 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 be more and more reliable.

REFERENCES

[1] OSHA Technical Guidance Manual. Section IV: Chapter 2. http://www.osha-slc.gov/dts/osta/otm/otm_iv/otm_iv_2.html

[2] McCabe, W. L.; Smith, J. C.; Harriott, P..(1986). Unit Operations of Chemical Engineering. McGraw-Hill International Edition.

[3] Fu, Limin.(1994). Neural Network in Computer Intelligence. McGraw-Hill International Edition.

[4] Anderson, D.; McNeil, G..(1992). Artificial Neural Networks Technology. Data & Analysis Center for Software.Rome. http://www.dacs.dtic.mil/techs/neural/neural.title.html

[5] Bose, N. K.; Liang, P..(1996). Neural Network Fundamentals with Graphics,Algorithms and Applications. McGraw-Hill International Edition.

[6] Antsalis, P. J.; Passins, K. M.. (1993). An Introduction to Intelligent and Autonomous Control. Norwell: Kluwer Academic Publishers.

[7] Fausett,L..(1994). Fundamentals of Neural Networks. New Jersey: Prentice-Hall.

[8] Demuth, H.; Beale, M.. (1998). Neural Network Toolbox for Use with MATLAB. Natick, MA: The Math Works Inc.

[9] Bhat, S.; Chatterjee, T.; Saraf, D.. (2003). On-line Data Processing and Product Properties Prediction for Crude Distillation Unit. AIChE 2003 Spring National meeting, New Orleans, Louisiana: Paper No. 115c.

[10] Bhat, N.; Minderman, P.; McAvoy, T.; Wong, N..(1990). Modelling Chemical Process Systems Via Neural Network Computation. IEEE Control System Magazine, April 10:24-31.

[11] Draeger, A.; Engell, S.; Ranke, H.. (1995). Model Predictive Control Using Neural Networks. IEEE Control System Magazine. 15:61-67

[12] Statistical Data. (2002). Tüpraş Oil Refinery, İzmit, Turkey.

[13] Saha, A..(1998). Neural Network Models in Excel for Prediction and Classification.
http://www.geocities.com/adotsaha/NNinExcel.html

[14] Edited by Zilouchian, A.; Jamshidi, M.. (2001). Intelligent Control Systems Using Soft Computing Methodologies. Chapter 7: Application of Neural Networks in Oil Refineries, CRC.

[15] Abiyev, R.H..(1993) Synthesis of Neural Network Structure For Neural Controller, Defining Quality of Products//Uchenie zapiski, AzGNA, N3, 103-106 pp, Baku. (Russian)

[16] Editors, Lastovkina, G.A.; Radchenko, M.G.; Radchenko, E.D.. (1986). Reference Book for Oil Specialist. Ximya, Leningrad (Sank-Petersburg). (Russian)

APPENDIX I

Collected Statistical Data (Tüpraş Oil Refinery, in İzmit, Turkey)

DATE	INPUTS							OUTPUT
	T_C (°C)	D_C (API)	F_C (m³/day)	F_G (m³/day)	T_T (°C)	D_K (API)	F_K (m³/day)	T_N (°C)
01.01.2002	321	34,1	13000	11493	140	45,9	1400	194
02.01.2002	321	35,2	12200	9661	145	45,6	1400	196
03.01.2002	321	35,6	11600	9141	137	45,4	1450	193
04.01.2002	320	35,2	12200	9450	135	45,6	1400	192
05.01.2002	323	34,8	12500	8923	151	44,6	1400	201
06.01.2002	323	31,1	12800	7940	144	44,4	1200	199
07.01.2002	322	36,2	11500	10399	143	44,6	1450	195
08.01.2002	321	35,4	11700	8500	143	44,9	1100	195
09.01.2002	320	31,3	12500	9190	135	43,9	1350	192
10.01.2002	320	30,6	12400	8873	136	44,3	1400	193
11.01.2002	326	34,8	12500	9181	159	44,6	1350	206
12.01.2002	325	32,1	12500	8590	163	44,3	1600	206
13.01.2002	324	34,9	12400	8876	159	44,6	1500	205
14.01.2002	327	30,6	12400	8898	151	44,3	1550	205
15.01.2002	324	35,4	12200	9450	145	45,5	1400	199
16.01.2002	320	34,8	12500	8590	137	44,6	1600	193
17.01.2002	326	36,2	11900	9670	152	46,1	1500	204
18.01.2002	327	31,7	12350	8736	154	44,4	1500	208
19.01.2002	321	34,8	12500	9870	143	44,5	1300	195
20.01.2002	322	31,1	12300	8876	145	44,2	1350	199
21.01.2002	321	35,6	11700	8500	139	45,4	1100	194
22.01.2002	320	33,3	12500	9783	141	44,8	1650	195
23.01.2002	324	32,5	12400	8898	145	44,2	1550	199
24.01.2002	326	34,8	12400	9358	157	44,5	1450	206
25.01.2002	320	35,3	11700	8500	137	45,6	1100	192
26.01.2002	320	31,6	12400	8876	136	44,7	1500	194
27.01.2002	325	33,8	12000	9672	146	44,6	1650	202
28.01.2002	323	34,8	12500	8590	148	44,5	1600	201
29.01.2002	321	35,6	11600	9141	136	45,4	1450	193
30.01.2002	328	34,8	12400	8898	155	44,6	1550	206
31.01.2002	328	34	12500	9783	154	45,7	1650	207
01.02.2002	321	34,1	12500	9885	140	45,9	1600	196
02.02.2002	320	35,2	12200	9450	135	45,6	1400	193
03.02.2002	320	34,8	12400	8876	136	44,6	1500	193
04.02.2002	329	33,3	12500	9783	155	44,8	1650	207
05.02.2002	321	35,3	11700	8500	138	45,6	1100	195
06.02.2002	325	31,7	12350	8736	157	44,4	1500	204
07.02.2002	324	36,4	11900	9670	145	45,1	1500	201
08.02.2002	321	35,6	11600	9141	144	45,4	1400	196
09.02.2002	323	34,8	12500	9885	153	44,5	1600	202
10.02.2002	320	31,6	12400	8898	136	44,7	1550	192
11.02.2002	327	30,6	12400	9358	150	44,3	1450	208
12.02.2002	326	36,6	12100	9391	154	45,1	1100	205

13.02.2002	323	35,4	11700	8998	151	44,9	1250	200
14.02.2002	324	31,6	12400	9560	148	44,7	1350	198
15.02.2002	322	35,2	12500	9133	144	45,6	1450	196
16.02.2002	321	35,6	11600	9141	141	45,4	1450	194
17.02.2002	320	35,7	12500	9133	140	45,4	1450	195
18.02.2002	325	36,6	12000	10359	163	44,9	1700	205
19.02.2002	326	30,8	12500	9190	157	44,9	1350	206
20.02.2002	327	30,8	12600	9808	153	44,9	1400	207
21.02.2002	322	36,2	12100	9513	143	46,1	1750	199
22.02.2002	326	36,4	11900	9800	152	45,1	1600	205
23.02.2002	320	32,1	12400	8898	141	44,3	1550	194
24.02.2002	321	36,2	11500	10399	142	44,6	1450	194
25.02.2002	320	34,8	12500	9870	137	44,5	1300	192
26.02.2002	323	35,2	12100	9391	146	45,8	1100	199
27.02.2002	325	31,1	12400	9358	158	43,9	1450	207
28.02.2002	324	32	12400	9560	154	45,3	1350	204
01.03.2002	326	36,2	11900	9850	153	45,5	1600	205
02.03.2002	329	32,5	12500	9783	153	44,2	1650	207
03.03.2002	322	31,1	12800	7940	146	44,2	1200	199
04.03.2002	321	36,6	12000	9672	143	45,1	1650	196
05.03.2002	320	36,6	12000	9672	137	44,9	1650	193
06.03.2002	325	33,3	12500	9885	151	44,8	1600	205
07.03.2002	326	34	13000	11493	156	45,7	1400	203
08.03.2002	328	35,2	12200	9450	153	45,6	1400	205
09.03.2002	321	33,8	12500	8890	145	44,7	1400	195
10.03.2002	325	33,8	12000	10359	160	44,7	1700	205
11.03.2002	326	34,9	12500	8923	158	44,6	1400	205
12.03.2002	322	30,8	12600	9808	145	44,9	1400	198
13.03.2002	323	35,7	12500	9190	151	45,4	1350	201
14.03.2002	323	35,4	12200	9661	154	44,9	1400	203
15.03.2002	321	30,8	12400	9358	136	44,9	1450	193
16.03.2002	320	31,7	12350	8776	138	44,4	1500	192
17.03.2002	324	35,2	12100	9513	144	45,8	1750	201
18.03.2002	324	36,4	11500	10400	145	45,1	1450	200
19.03.2002	327	30,6	12400	8873	153	44,3	1400	208
20.03.2002	325	36,2	12100	10271	149	45,5	1750	202
21.03.2002	324	32,5	12500	9870	149	44,2	1300	202
22.03.2002	321	30,6	12500	9783	141	44,3	1650	195
23.03.2002	322	35,2	12400	8898	147	45,6	1550	199
24.03.2002	320	31,3	12500	9885	139	43,9	1600	193
25.03.2002	326	34,1	13000	11493	158	45,9	1700	205
26.03.2002	323	31,1	12800	7900	152	44,2	1200	201
27.03.2002	323	35,2	12100	9391	151	45,8	1100	202
28.03.2002	325	31,7	12400	9560	147	44,4	1350	204
29.03.2002	324	35,6	11600	9141	145	45,4	1450	202
30.03.2002	321	33,3	12400	8873	144	44,8	1400	195
31.03.2002	325	36,2	11900	9800	154	44,6	1500	204
01.04.2002	321	33,8	12400	8876	142	44,7	1500	193
02.04.2002	323	36,6	12000	9672	153	45,1	1650	202
03.04.2002	329	31,3	12300	9840	156	43,9	1400	208
04.04.2002	328	33,3	12500	9133	153	44,8	1450	207
05.04.2002	326	34,8	12500	9190	156	44,5	1350	205
06.04.2002	324	32,5	12400	9358	144	44,2	1450	199

07.04.2002	321	35,6	11600	9141	145	45,4	1450	195
08.04.2002	324	35,3	11700	8500	145	45,6	1100	198
09.04.2002	322	35,2	12100	9391	147	45,8	1100	196
10.04.2002	324	34,9	12400	9560	159	44,6	1350	205
11.04.2002	326	34,8	12500	9133	159	44,5	1450	205
12.04.2002	325	36,2	12100	9513	163	44,5	1750	205
13.04.2002	324	36,6	12000	10359	157	44,9	1700	204
14.04.2002	325	36,2	11900	9800	157	44,5	1500	205
15.04.2002	326	36,2	11900	9850	152	44,6	1600	204
16.04.2002	326	32,1	12500	8923	152	44,3	1400	205
17.04.2002	329	31,3	12500	9181	153	43,9	1350	207
18.04.2002	326	31,1	12400	8876	154	44,4	1500	206
19.04.2002	327	30,6	12400	8998	150	44,3	1550	208
20.04.2002	324	31,6	12400	9358	149	44,7	1450	202
21.04.2002	324	32,5	12500	9870	153	44,2	1300	205
22.04.2002	323	31,1	12300	9889	153	43,9	1400	200
23.04.2002	324	32	12300	8876	146	45,3	1350	202
24.04.2002	325	31,7	12350	8736	146	44,4	1500	202
25.04.2002	325	33,8	12500	9783	151	44,6	1650	203
26.04.2002	324	33,8	12000	9672	141	44,7	1650	201
27.04.2002	323	35,7	12500	9885	140	45,4	1600	196
28.04.2002	321	34,1	13000	11493	141	45,9	1400	194
29.04.2002	322	34	13050	9839	140	45,7	1350	196
30.04.2002	322	31,1	12800	7940	142	44,2	1200	196

APPENDIX II

Model results in NNinExcel for 10 neurons in single hidden layer

Neural Network Model for Prediction Created On : 12-May-05

MSE(Training) 0,957 MSE(Validation) 1,5452

Number of Hidden Layers 1

Layer Sizes 7 10 0 1

Slope 0,0625

Model Output (Untransformed) 0,0001

ABS((Tru - Predicted) / Tru) #DIV/0!

Output - Predicted by the model

				Cont	Cont	Cont	Cont	Cont
	Bias	:ru		ıde Fow	ìas Oil Flo	Top Temp	rosene De	rosene Flow
Slope	1,0000			0,0006	0,0003	0,0357	0,4646	0,0015

Enter your Inputs in the range AG115:AM115 - the cells marked in green.

	Bias	Temp.	Density	Crude Fow	Gas Oil Flow	Top Temp.	Kerosene Dens.	Kerosene Flow			
Transformed Input	1	-35,5556	-5,1000	-7,4194	-2,1987	-4,8214	-19,9546	-1,6923			
Hdn1_bia	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000			
Hdn1_Nrr	-0,5359	0,5546	0,2333	-0,4629	-0,2082	-0,1225	-0,2810	0,5294			
Hdn1_Nrr	0,1827	-1,1270	0,4465	0,5780	0,4744	-0,9145	0,3461	-0,5055			
Hdn1_Nrr	-0,3331	-0,6168	0,0631	-0,1587	0,5217	-0,4849	-0,3123	-0,2134			
Hdn1_Nrr	-0,6024	1,3975	-0,5890	-0,3848	-0,3381	1,2977	-0,3777	-0,4471	-41,6521		
Hdn1_Nrr	0,2580	-0,5661	0,2858	-0,4606	-0,1310	-0,0363	-0,3136	0,0174	28,9645		
Hdn1_Nrr	0,1821	-1,4602	-0,3215	-0,4739	0,4463	-0,9447	-0,3884	-0,2217	68,6003		
Hdn1_Nrr	0,1026	0,4792	0,0791	0,2475	-0,2329	-0,0359	0,3924	-0,0343	-26,2616		
Hdn1_Nrr	0,7012	-1,7219	-0,1074	-0,1365	-0,2937	-1,9870	-0,5337	-0,6667	85,4898		
Hdn1_Nrr	-0,0719	-1,0238	-0,2946	-0,4608	0,3983	-0,8882	-0,5048	0,1365	54,4317		
Hdn1_Nrr	-0,2156	1,6436	-0,2616	-0,2978	-0,0926	1,4475	0,4740	-0,0040	-71,3400		
	1,0000	0,0000	1,0000	1,0000	0,0000	1,0000	1,0000	0,0000	1,0000	1,0000	0,0000
Op_bias	*0,0000*	*0,0000*	*0,0000*	*0,0000*	*0,0000*	*0,0000*	*0,0000*	*0,0000*	*0,0000*	*0,0000*	*0,0000*
Op_Nrn1	0,0010	0,4458	-1,8721	-0,9107	2,1307	-0,4229	-1,7630	0,2717	-2,7243	-1,2301	2,1551
	1,0000	0,0001									

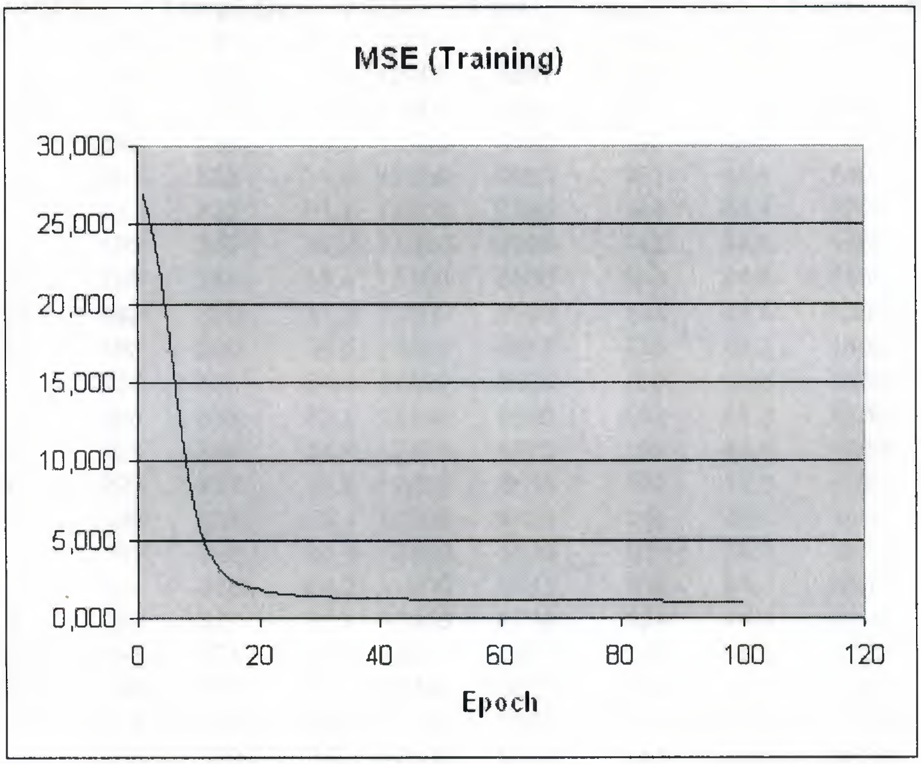
APPENDIX III

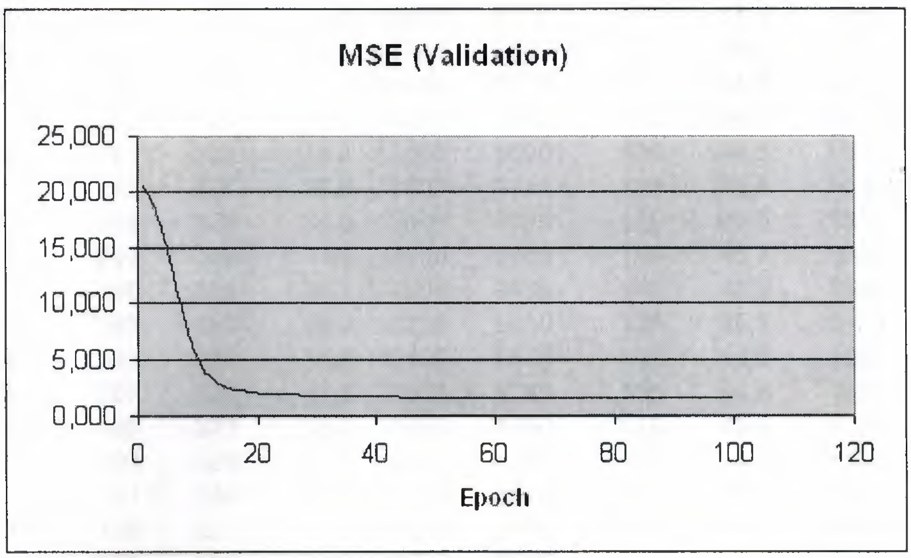
Simulation Results for Neural Network Model in NNinExcel with 10 Neurons in one Hidden Layer

	Avg. error per Input (Original Scale) (Training Set)		Avg. error per Input (Original Scale) (Validation Set)	
Epoch	MSE (Original Scale)	ARE (%)	MSE (Original Scale)	ARE (%)
1	26,982	2,31%	20,593	2,02%
2	25,531	2,25%	19,615	1,97%
3	23,815	2,17%	18,398	1,91%
4	21,645	2,07%	16,815	1,83%
5	18,988	1,93%	14,848	1,73%
6	15,997	1,77%	12,605	1,59%
7	12,968	1,59%	10,301	1,44%
8	10,225	1,41%	8,193	1,28%
9	7,973	1,24%	6,460	1,13%
10	6,252	1,09%	5,150	0,99%
11	4,988	0,96%	4,210	0,87%
12	4,076	0,86%	3,552	0,77%
13	3,419	0,77%	3,095	0,70%
14	2,941	0,71%	2,776	0,65%
15	2,590	0,66%	2,551	0,62%
16	2,328	0,62%	2,388	0,60%
17	2,130	0,59%	2,270	0,59%
18	1,977	0,57%	2,181	0,58%
19	1,857	0,55%	2,113	0,57%
20	1,762	0,53%	2,060	0,57%
21	1,685	0,52%	2,017	0,57%
22	1,622	0,51%	1,983	0,57%
23	1,571	0,50%	1,954	0,57%
24	1,527	0,50%	1,929	0,57%
25	1,490	0,49%	1,908	0,57%
26	1,458	0,49%	1,889	0,57%
27	1,430	0,48%	1,872	0,57%
28	1,406	0,48%	1,857	0,57%
29	1,384	0,47%	1,843	0,56%
30	1,365	0,47%	1,829	0,56%
31	1,347	0,47%	1,817	0,56%
32	1,331	0,46%	1,805	0,56%
33	1,317	0,46%	1,795	0,56%
34	1,303	0,46%	1,784	0,56%
35	1,291	0,46%	1,774	0,56%
36	1,280	0,45%	1,765	0,55%
37	1,269	0,45%	1,756	0,55%

38	1,259	0,45%	1,747	0,55%
39	1,250	0,45%	1,739	0,55%
40	1,242	0,44%	1,731	0,55%
41	1,233	0,44%	1,723	0,55%
42	1,226	0,44%	1,716	0,55%
43	1,218	0,44%	1,709	0,55%
44	1,212	0,44%	1,702	0,54%
45	1,205	0,43%	1,696	0,54%
46	1,199	0,43%	1,690	0,54%
47	1,193	0,43%	1,684	0,54%
48	1,188	0,43%	1,678	0,54%
49	1,182	0,43%	1,673	0,54%
50	1,177	0,43%	1,667	0,54%
51	1,172	0,42%	1,662	0,53%
52	1,168	0,42%	1,658	0,53%
53	1,163	0,42%	1,653	0,53%
54	1,159	0,42%	1,649	0,53%
55	1,155	0,42%	1,644	0,53%
56	1,151	0,42%	1,640	0,53%
57	1,148	0,42%	1,636	0,53%
58	1,144	0,41%	1,632	0,53%
59	1,141	0,41%	1,629	0,53%
60	1,137	0,41%	1,625	0,52%
61	1,134	0,41%	1,622	0,52%
62	1,131	0,41%	1,619	0,52%
63	1,128	0,41%	1,616	0,52%
64	1,125	0,41%	1,613	0,52%
65	1,123	0,41%	1,610	0,52%
66	1,120	0,41%	1,607	0,52%
67	1,118	0,41%	1,605	0,52%
68	1,115	0,41%	1,602	0,52%
69	1,113	0,41%	1,600	0,52%
70	1,110	0,40%	1,597	0,52%
71	1,108	0,40%	1,595	0,52%
72	1,106	0,40%	1,593	0,52%
73	1,104	0,40%	1,591	0,52%
74	1,102	0,40%	1,587	0,51%
75	1,100	0,40%	1,585	0,51%
76	1,098	0,40%	1,582	0,51%
77	1,096	0,40%	1,580	0,51%
78	1,094	0,40%	1,579	0,51%
79	1,092	0,40%	1,575	0,51%
80	1,091	0,40%	1,571	0,51%
81	1,089	0,40%	1,570	0,51%
82	1,087	0,40%	1,568	0,51%
83	1,086	0,40%	1,565	0,51%
84	1,084	0,40%	1,563	0,51%
85	1,083	0,40%	1,563	0,51%
86	1,081	0,40%	1,560	0,51%
87	1,080	0,40%	1,559	0,51%
88	1,078	0,40%	1,557	0,51%
89	1,077	0,40%	1,557	0,51%
90	1,075	0,40%	1,555	0,51%

91	1,074	0,40%	1,551	0,51%
92	1,072	0,40%	1,549	0,51%
93	1,071	0,40%	1,547	0,51%
94	1,070	0,40%	1,543	0,51%
95	1,069	0,40%	1,537	0,50%
96	1,067	0,40%	1,533	0,50%
97	1,066	0,39%	1,531	0,50%
98	1,065	0,39%	1,531	0,50%
99	1,064	0,39%	1,530	0,49%
100	1,063	0,39%	1,529	0,49%





APPENDIX IV

Training and Verification Results of Neural Network Using NNinExcel

(The best model architecture with 10 neurons in single hidden layer)

Train / Valid	Obs No.	Output Actual Naphtha 95 %	Cont Crude Temp.	Cont Crude Density	Cont Crude Flow	Cont Gas O. Flow	Cont Top Temp.	Cont Kero. Dens.	Cont Kero. Flow	Predicted Naphtha 95 %
1	1	194	321	34,1	13000	11493	140	45,9	1400	194,2736
1	2	196	321	35,2	12200	9661	145	45,6	1400	196,2115
1	3	193	321	35,6	11600	9141	137	45,4	1450	194,0851
1	4	192	320	35,2	12200	9450	135	45,6	1400	193,0749
1	5	201	323	34,8	12500	8923	151	44,6	1400	201,3146
1	6	199	323	31,1	12800	7940	144	44,4	1200	198,998
1	7	195	322	36,2	11500	10399	143	44,6	1450	195,7518
1	8	195	321	35,4	11700	8500	143	44,9	1100	194,9353
1	9	192	320	31,3	12500	9190	135	43,9	1350	192,6914
1	10	193	320	30,6	12400	8873	136	44,3	1400	193,02
1	11	206	326	34,8	12500	9181	159	44,6	1350	206,0169
1	12	206	325	32,1	12500	8590	163	44,3	1600	206,5603
1	13	205	324	34,9	12400	8876	159	44,6	1500	205,0416
1	14	205	327	30,6	12400	8898	151	44,3	1550	206,1703
1	15	199	324	35,4	12200	9450	145	45,5	1400	200,5706
1	16	193	320	34,8	12500	8590	137	44,6	1600	193,3568
1	17	204	326	36,2	11900	9670	152	46,1	1500	204,9627
1	18	208	327	31,7	12350	8736	154	44,4	1500	206,4378
1	19	195	321	34,8	12500	9870	143	44,5	1300	194,4635
1	20	199	322	31,1	12300	8876	145	44,2	1350	197,6479
1	21	194	321	35,6	11700	8500	139	45,4	1100	194,1159
1	22	195	320	33,3	12500	9783	141	44,8	1650	194,0831
1	23	199	324	32,5	12400	8898	145	44,2	1550	201,1415
1	24	206	326	34,8	12400	9358	157	44,5	1450	205,7763
1	25	192	320	35,3	11700	8500	137	45,6	1100	193,1841
1	26	194	320	31,6	12400	8876	136	44,7	1500	193,2187
1	27	202	325	33,8	12000	9672	146	44,6	1650	202,768
1	28	201	323	34,8	12500	8590	148	44,5	1600	200,7042
1	29	193	321	35,6	11600	9141	136	45,4	1450	193,8643
1	30	206	328	34,8	12400	8898	155	44,6	1550	206,5662
1	31	207	328	34	12500	9783	154	45,7	1650	206,3591
1	32	196	321	34,1	12500	9885	140	45,9	1600	195,2686
1	33	193	320	35,2	12200	9450	135	45,6	1400	193,0749
1	34	193	320	34,8	12400	8876	136	44,6	1500	193,0526
1	35	207	329	33,3	12500	9783	155	44,8	1650	206,8209
1	36	195	321	35,3	11700	8500	138	45,6	1100	194,0199
1	37	204	325	31,7	12350	8736	157	44,4	1500	205,8881
1	38	201	324	36,4	11900	9670	145	45,1	1500	200,3221
1	39	196	321	35,6	11600	9141	144	45,4	1400	196,0828
1	40	202	323	34,8	12500	9885	153	44,5	1600	201,7525

1	41	192	320	31,6	12400	8898	136	44,7	1550	193,2736
1	42	208	327	30,6	12400	9358	150	44,3	1450	205,8429
1	43	205	326	36,6	12100	9391	154	45,1	1100	204,856
1	44	200	323	35,4	11700	8998	151	44,9	1250	201,4788
1	45	198	324	31,6	12400	9560	148	44,7	1350	202,0461
1	46	196	322	35,2	12500	9133	144	45,6	1450	197,4933
1	47	194	321	35,6	11600	9141	141	45,4	1450	195,1779
1	48	195	320	35,7	12500	9133	140	45,4	1450	193,8885
1	49	205	325	36,6	12000	10359	163	44,9	1700	205,7776
1	50	206	326	30,8	12500	9190	157	44,9	1350	206,272
1	51	207	327	30,8	12600	9808	153	44,9	1400	206,0752
1	52	199	322	36,2	12100	9513	143	46,1	1750	198,0521
1	53	205	326	36,4	11900	9800	152	45,1	1600	204,8549
1	54	194	320	32,1	12400	8898	141	44,3	1550	194,014
1	55	194	321	36,2	11500	10399	142	44,6	1450	194,356
1	56	192	320	34,8	12500	9870	137	44,5	1300	192,8147
1	57	199	323	35,2	12100	9391	146	45,8	1100	199,0869
1	58	207	325	31,1	12400	9358	158	43,9	1450	205,7965
1	59	204	324	32	12400	9560	154	45,3	1350	204,157
1	60	205	326	36,2	11900	9850	153	45,5	1600	205,1008
1	61	207	329	32,5	12500	9783	153	44,2	1650	206,7367
1	62	199	322	31,1	12800	7940	146	44,2	1200	198,0663
1	63	196	321	36,6	12000	9672	143	45,1	1650	195,5921
1	64	193	320	36,6	12000	9672	137	44,9	1650	193,2762
1	65	205	325	33,3	12500	9885	151	44,8	1600	203,9854
1	66	203	326	34	13000	11493	156	45,7	1400	204,6291
1	67	205	328	35,2	12200	9450	153	45,6	1400	206,2206
1	68	195	321	33,8	12500	8890	145	44,7	1400	196,0433
1	69	205	325	33,8	12000	10359	160	44,7	1700	205,7573
1	70	205	326	34,9	12500	8923	158	44,6	1400	205,9738
1	71	198	322	30,8	12600	9808	145	44,9	1400	197,7053
1	72	201	323	35,7	12500	9190	151	45,4	1350	201,1994
1	73	203	323	35,4	12200	9661	154	44,9	1400	202,1432
1	74	193	321	30,8	12400	9358	136	44,9	1450	193,8136
1	75	192	320	31,7	12350	8776	138	44,4	1500	193,4408
1	76	201	324	35,2	12100	9513	144	45,8	1750	201,2248
1	77	200	324	36,4	11500	10400	145	45,1	1450	199,9025
1	78	208	327	30,6	12400	8873	153	44,3	1400	206,339
1	79	202	325	36,2	12100	10271	149	45,5	1750	203,1449
1	80	202	324	32,5	12500	9870	149	44,2	1300	201,5067
1	81	195	321	30,6	12500	9783	141	44,3	1650	194,9927
1	82	199	322	35,2	12400	8898	147	45,6	1550	199,1691
1	83	193	320	31,3	12500	9885	139	43,9	1600	193,2939
1	84	205	326	34,1	13000	11493	158	45,9	1700	205,1467
1	85	201	323	31,1	12800	7900	152	44,2	1200	202,5703
1	86	202	323	35,2	12100	9391	151	45,8	1100	201,1709
1	87	204	325	31,7	12400	9560	147	44,4	1350	202,8615
1	88	202	324	35,6	11600	9141	145	45,4	1450	201,1877
1	89	195	321	33,3	12400	8873	144	44,8	1400	195,881
1	90	204	325	36,2	11900	9800	154	44,6	1500	204,3318
0	91	193	321	33,8	12400	8876	142	44,7	1500	195,2708
0	92	202	323	36,6	12000	9672	153	45,1	1650	202,1239
0	93	208	329	31,3	12300	9840	156	43,9	1400	206,9636

0	94	207	328	33,3	12500	9133	153	44,8	1450	206,4294
0	95	205	326	34,8	12500	9190	156	44,5	1350	205,5767
0	96	199	324	32,5	12400	9358	144	44,2	1450	200,0956
0	97	195	321	35,6	11600	9141	145	45,4	1450	196,5945
0	98	198	324	35,3	11700	8500	145	45,6	1100	200,9119
0	99	196	322	35,2	12100	9391	147	45,8	1100	197,9821
0	100	205	324	34,9	12400	9560	159	44,6	1350	204,6017
0	101	205	326	34,8	12500	9133	159	44,5	1450	206,0732
0	102	205	325	36,2	12100	9513	163	44,5	1750	206,06
0	103	204	324	36,6	12000	10359	157	44,9	1700	203,9623
0	104	205	325	36,2	11900	9800	157	44,5	1500	204,9784
0	105	204	326	36,2	11900	9850	152	44,6	1600	204,7972
0	106	205	326	32,1	12500	8923	152	44,3	1400	205,4215
0	107	207	329	31,3	12500	9181	153	43,9	1350	206,8278
0	108	206	326	31,1	12400	8876	154	44,4	1500	206,0097
0	109	208	327	30,6	12400	8998	150	44,3	1550	206,0113
0	110	202	324	31,6	12400	9358	149	44,7	1450	202,7592
0	111	205	324	32,5	12500	9870	153	44,2	1300	203,0443
0	112	200	323	31,1	12300	9889	153	43,9	1400	202,1601
0	113	202	324	32	12300	8876	146	45,3	1350	202,0194
0	114	202	325	31,7	12350	8736	146	44,4	1500	203,3211
0	115	203	325	33,8	12500	9783	151	44,6	1650	203,9304
0	116	201	324	33,8	12000	9672	141	44,7	1650	199,3191
0	117	196	323	35,7	12500	9885	140	45,4	1600	197,0366
0	118	194	321	34,1	13000	11493	141	45,9	1400	194,4951
0	119	196	322	34	13050	9839	140	45,7	1350	195,5655
0	120	196	322	31,1	12800	7940	142	44,2	1200	196,2349