DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

COLOR RECIPE PREDICTION WITH NEURAL NETWORKS

by

Mehmet Volkan SAĞIRLIBAŞ

September, 2009

İZMİR

COLOR RECIPE PREDICTION WITH NEURAL NETWORKS

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science
in Electrical & Electronics Engineering Program

by

Mehmet Volkan SAĞIRLIBAŞ

September, 2009

İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled "**COLOR RECIPE PREDICTION WITH NEURAL NETWORKS**" completed by **MEHMET VOLKAN SAĞIRLIBAŞ** under supervision of **ASSIST. PROF. DR. YAVUZ ŞENOL** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Yavuz ŞENOL

Supervisor

Prof. Dr. Mustafa GÜNDÜZALP

(Jury Member)

Assoc. Prof. Dr. Merih SARIIŞIK

(Jury Member)

Prof. Dr. Cahit HELVACI

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

I would like to thank to my supervisor Asst. Prof. Dr. Yavuz ŞENOL for his valuable guidance and support during the course of this thesis.

I am grateful to Assoc. Prof. Dr. Merih SARIIŞIK from Textile Engineering department of Dokuz Eylül University for her support and help in the part of textile in this thesis. Also I would like to thank to Ekoten A.Ş. for their help for supplying the data in this thesis.

I would like to thank to my parents for their loving care and continuous encouragement.

I wish to express my special thanks to my wife Devin SAĞIRLIBAŞ for her endless support, love, care and especially patience.

Foremost, my special thanks are for my son Sarp SAĞIRLIBAŞ for his joining to my life in these exhausting times and calming me.

Mehmet Volkan SAĞIRLIBAŞ

COLOR RECIPE PREDICTION WITH NEURAL NETWORKS

ABSTRACT

The textile is colored most of the time for giving a more attractive appearance or effect. In these colorings, in order to have exactly the same color; color recipe is used for the textile that wanted to be the same color but dyed in different times and with different dyeing machines. Because of this, color recipe prediction has a very important place in the textile industry. Computerized color measurement devices used in dye houses are important devices for color recipe prediction. Predicting the color recipe correctly increases the dyeing performance; decreases complete dyeing process time and decrease the possible errors that are likely to be made.

There are a lot of methods used in color recipe prediction. However, because of the nonlinear structure of the color recipes, in this thesis color recipe prediction has been performed by using artificial neural networks and fuzzy logic. While making these predictions, the used data groups made according to CIE system (Lab, Lch, XYZ) and reflectance values. With various programs; radial basis function neural network (RBF NN), feed-forward multilayer perceptron neural network (MLP NN) and fuzzy logic developed in MATLAB were used for calculating the color recipe and their results were compared in detail.

In the applications with these three methods it was seen that as the data quantity for training was increased from 250 to 400, the error percentage of the system decreased to 0%. This shows that the quantity of training data is very important for successful training of the system.

As a result in all three methods the success of 100% was achieved, however RBF was the most successful method compared to MLP and fuzzy logic. RBF was both faster and was also more stable than the others. It reaches to 0% error value faster.

Keywords: Artificial Neural Networks, Fuzzy Logic, Radial Basis Function, Multi Layer Perceptron, Color Recipe Prediction

SİNİR AĞLARI İLE RENK REÇETESİ TAHMİNİ

ÖZ

Kumaşlar çoğu zaman daha çekici bir görünüm veya etki vermek için renklendirilirler. Bu renklendirmelerde aynı renkte olması istenen ve farklı zamanlarda ve farklı makinelerle boyanan kumaşlarda istenen rengin birebir tutturulması için renk reçetesi kullanılmaktadır. Dolayısıyla tekstil alanında renk reçetesi tahmini çok önemli bir yer tutmaktadır. Boyahanelerde bulunan bilgisayarlı renk ölçüm cihazları renk reçetesi tahmininde kullanılan önemli cihazlardır. Renk reçetesinin doğru tahmin edilmesi, performansı arttırmakta, süreci kısaltmakta ve olası hataları azaltmaktadır.

Renk reçetesi tahmininde kullanılan birçok yöntem bulunmaktadır. Ancak, renk reçeteleri doğrusal olmayan bir yapıya sahip olduğu için bu tezde yapay sinir ağları ve bulanık mantık yöntemleri ile renk reçetesi tahmin edilmeye çalışılmıştır. Bu tahminler yapılırken CIE sistemi (Lab, Lch, XYZ) ve reflektans değerlerine göre oluşturulmuş veri grupları kullanılmıştır. MATLAB de geliştirilen çeşitli programlarla radyal tabanlı fonksiyon sinir ağı, feed-forward çok katmanlı perseptron sinir ağı ve bulanık mantık ile reçete tahmini yapılmış ve sonuçları detaylı olarak karşılaştırılmıştır.

Bu üç metot ile yapılan uygulamalarda görüldü ki, eğitim için ayrılan veri miktarı 250’ den 400’ e çıkarılınca sistemin hata oranı %0’ a düştü. Bu gösterir ki, eğitim için kullanılan veri miktarı sistemin başarısı için çok önemlidir.

Sonuç olarak bu üç metotta da %100 başarı oranı elde edilmiştir, ama RBF, MLP ve bulanık mantığa göre daha başarılı bir metot olmuştur. RBF öteki metotlara göre hem daha hızlı hem de daha kararlıdır. RBF, %0 hata oranına daha çabuk ulaşır.

Anahtar Kelimeler: Yapay Sinir Ağları, Bulanık Mantık, Radyal Tabanlı Fonksiyon, Çok Katmanlı Perseptron, Renk Reçetesi Tahmini

CONTENTS

Page

CHAPTER ONE

INTRODUCTION

Color recipe prediction has a very important place in the textile industry. Because every customer wants to have the same color of textile in every repetitively given order, even if they are ordered in different times.

Normally, it is a very difficult process to have exactly the same color in the textiles dyed in different times. In workshops the experienced colorists do this process by their visual experiences, but because of the personal properties, it may not have so stable results, because when the colorist changes, because of the change of the experience, the quality might change. Therefore, the important part of the recipe depends on the performance of the experts.

For this reason, color recipe prediction is one of the most important problems in the textile industry and the professionals of this subject try to carry this process in a stable platform that does not change with personal properties that the people do the process.

Accordingly, over the years, some computer based solutions were developed. Although they are more stable than the human experience, there are some bottlenecks in the studies because most of the color recipes have a nonlinear structure. Therefore, it is not an easy process to predict the recipe.

For this reason, in this thesis artificial neural network and fuzzy logic were used separately in order to have a system that predicts color recipe automatically. These two methods were selected because of non-linear structures. Besides, when they were once trained, whatever data related to that recipe given to the system, they are expected to give correct response as part of the generalization property.

This chapter is an introduction that is related to the general explanation of the thesis. In the second, third and fourth chapters, theoretical background about textile, artificial neural networks and fuzzy logic are given, respectively. The fifth chapter is

about the applications of neural networks and fuzzy logic and their comparison with the graphs. Finally a conclusion part is given.

CHAPTER TWO

TEXTILE BACKGROUND

2.1 Color

Color is a perception that occurs by receiving the different wavelengths of light by the retina of the eye. This perception has a variety that comes from the amounts of light absorption, reflection or emission of different materials. As a result of this variety; different colors and shades are formed.

The portion of electromagnetic spectrum that is visible to the human eye is called the visible spectrum. This means that; this spectrum can be detected by the human eye. Electromagnetic radiation in the range of wavelengths that are detected by the human eye is called visible light or simply light. A typical human eye will respond to wavelengths from 350 to 700 nm as seen in Figure 2.1. (Starr C., Evers C., Starr L., 2005)

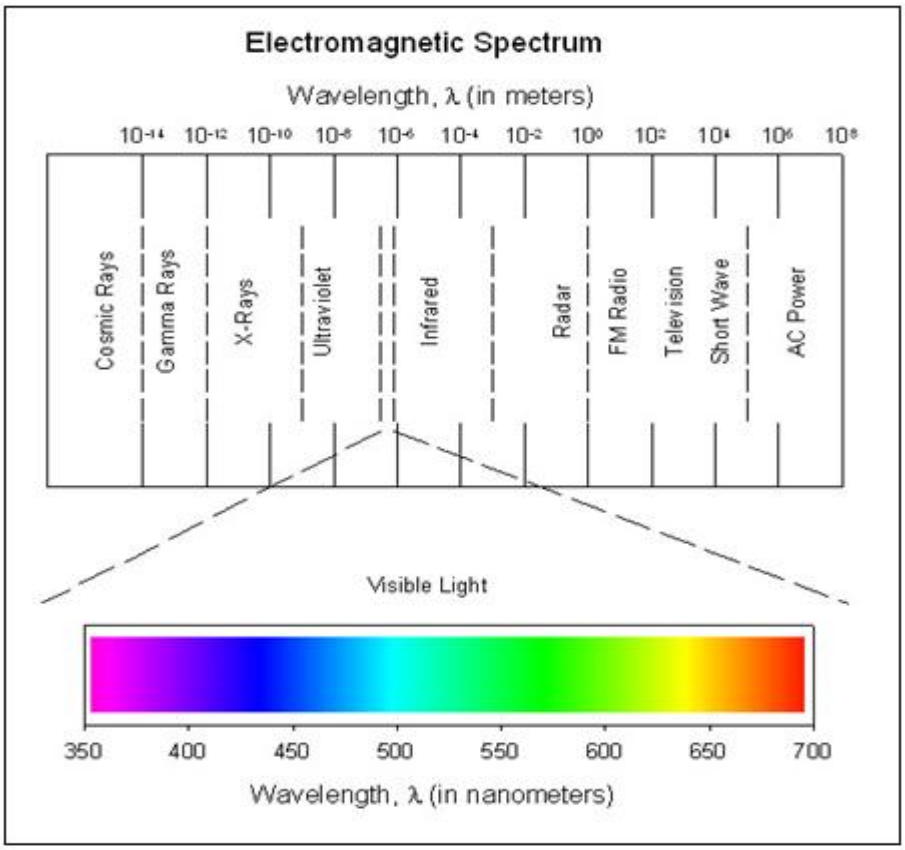

Figure 2.1 Electromagnetic Spectrum

2.2 Color Collections and Color Systems

From the existence of human being, people admire to the magnificent colors on the living creatures and nonliving structures. They started to collect the colors and preserve them as a whole. By this way, the color collections come into being. (Duran, 2001)

2.2.1 Color Collections

In the past and today, color examples obtained by dyeing are preserved as files. To keep the secrecy a code number is given to the color examples. The agreements between the company and the client are done by using these code numbers. A color collection prepared by this way is not only difficult but also expensive. The disadvantages of this color collection way are as follows:

- The appearance of a color is firstly depends on the objects' upper surface. For this reason with the same prescription, it is impossible to obtain the same colors in the fabrics that not match with the upper surface structure of the examples in the collection.
- The color collection consists of millions of colors. An experienced colorist can recognize the differences even between 5-10 millions of colors. But if a color that is not exactly the same any of the color in the collection, is tried to obtain by looking the other colors, a lot of difficulties can occur.

If making a color collection is inevitable than attention should be paid to these points:

- The examples in the color collection must be prepared in big numbers and very carefully.
- The type, upper surface structure and the other technical features of the material used in the color collection should suit to the fabric features that will be used in the practice.

- The obtained examples must be arranged according to their colors and they should be protected from light, moisture and dust. (Duran, 2001)

2.2.2 Color Systems

Until today there have been a lot of color system which has been found by scientists. These are; Dell Porta (1593), Aguilonious (1613), Kircher (1646), Neuston (1660), Waller (1686), Lambert (1772), Goether (1793),Runge (1810), Herschel (1817), Schreiber (1840), Maxwell (1872), Wundt (1874), Von Bezold (1876), Höfler (1833), Titchener (1887), Chevreul (1889), Wundt (1893), Ebbinghous (1902), Munsell (1905), Rood (1910), Munsell (1916), Ostwald (1917), Klee (1924), Boring (1929), Pope (1929), CIE (1931), Johanson (1939), Hickethier (1940), Rosch (1972), Gerritsen (1975). (Duran, 2001)

2.2.2.1 The Ostwald Color System

Ostwald system is probably the best known system based on the results of disk colorimetry. In an ideal system, colors in the Ostwald system are produced by light reflected from a spinning disk consisting of segments of white, black, and a high-chroma sample designated a full color. These full colors are enough to provide a circle of hues. On a page of the Ostwald system, colors are described by their full color content, white content, and black content. The Ostwald system organization emphasizes scales of colors having approximately constant hue, constant black content, and constant white content. It is particularly suitable to use for artists, painters, ink makers and others who work with mixtures of colored pigment with black and white pigments.

2.2.2.2 The Munsell Color System

Munsell System is perhaps the best known of all color-order systems. It is based on the guiding principle of equal visual perception. The Munsell System is both a collection of samples painted to represent equal intervals of visual perception

between adjacent samples, and a system for describing all possible colors in terms of its three coordinates, Munsell Hue, Munsell Value, and Munsell Chroma.

The coordinates of the Munsell Color System correspond to three variables commonly used to describe color; hue is that quality of color described by the words red, yellow, green, blue, and so forth; value is that quality by which a color can be c1assified as equivalent in lightness to some member of a series of gray samples ranging from white to black; chroma is the quality that describes the degree of difference between a color (which is itself not a white, gray, or black) and a gray of the same value or lightness.

The Munsell System has several outstanding features that contribute to its usefulness and wide acceptance. The first one is its conformance to equal visual perception. There is very little evidence for deviation from equal steps of perception in any of the Munsell coordinates within the limits of chroma (6 - 10) set by the samples of the original Munsell Book of Color.

A second major advantage that Munsell System has is its notation is not linked to or limited by existing samples. Any conceivable color can be fitted into the system, if it can be produced with existing colorants or not.

The other advantage of the Munsell System is that the samples of the Munsell book of Color are prepared to very close tolerances, so that the user can rely on the samples in his copy being very close in color to those from other copies or other modern editions of the Munsell Book of Color.

2.2.2.3 The Manfred Richter Color System

This system was published by Manfred Richter in 1950. This system was later accepted as DIN-color system. It is based upon the idea that colors are ordered along three subjective dimensions, i.e. hue, saturation, and brightness. In this system the colors that have certain brightness are gathered in the same group. (Duran, 2001)

2.2.2.4 The CIE Color System

The most important of these systems, which are usually used in connection with instruments for color measurement, is the CIE system (Commission International de l'Eclairage or International Commission on Illumination). The CIE introduced the element of standardization of source and observer in 1931, and the methodology to derive numbers that provide a measure of a color seen under a standard source of illumination by a standard observer.

In 1965 the CIE recommended a series of illuminants to supplement A, B and C, based on definitive studies of the spectral power distribution of natural daylight. They represent average daylight over spectral range 300-830 nm and have correlated color temperatures between 4000 and 25,000 K.

The CIE system characterizes colors by a luminance parameter Y and two color x and y which specify the point on the chromaticity diagram in the Figure 2.2. This system offers more precision in color measurement than do the Munsell and Ostwald systems because the parameters are based on the spectral power distribution (SPD) of the light emitted from a colored object and are factored by sensitivity curves which have been measured for the human eye.

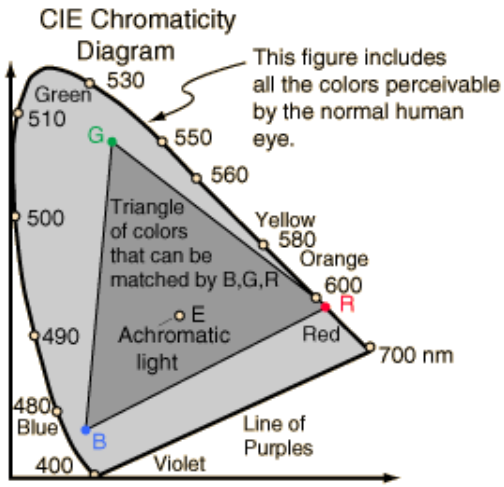

Figure 2.2 CIE Chromaticity diagram

2.2.2.5 The CIELab Color System

The CIE recommended the use of two visually uniform color systems considered best in 1976. One of these systems has a wide use in the textile industry: the CIELab system uses three coordinates L*, a* and b*- which can be calculated from the tristimulus values X, Y, and Z. To distinguish them from those of similar but different systems used even earlier is the purpose of the asterisk in these parameters. Figure 2.3 shows the a* and b* axes which are right angles and intersect at the neutral point (gray or white, depending on the lightness).

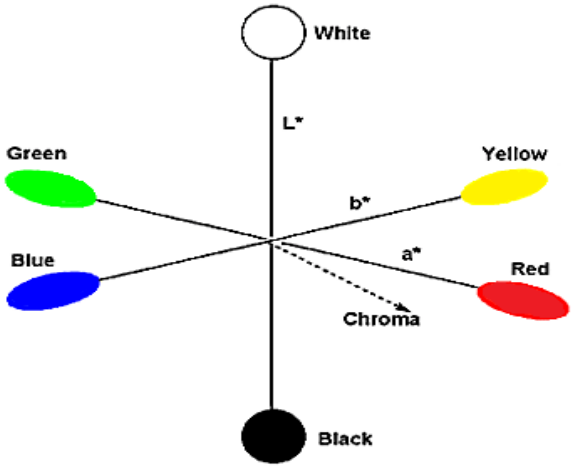

Figure 2.3 The CIELab color system

The third axis is called L*, a measure of lightness, which is perpendicular to the plane formed by a* and b*, and intersects it at the neutral point. Colors of the same hue lie on straight lines are running outward from the neutral point in the plane are formed by a* and b*. Here the angle of rotation h in degrees (increasing from red to yellow) is a measure of the hue. For example, h = 0º corresponds to a red shade, h = 90º to a yellow, and h = 270º to a blue. Chroma Cº represents the distance of the point from the neutral point and thus is a measure of the brilliance or clarity of the color at a given lightness.

A color can be represented by either in terms of the coordinates L*, a* and b*, or with L*, C* and h*. Generally, the coloristic way of thinking is probably more nearly approached by specification of hue angle h and chroma C* than by specification of coordinates a* and b*, which are more convenient for gray and dull

colors. L* represents a measure of the lightness of the color in any case, the values of L* vary from 0 for black to 100 for white, and the highest values of a and b for very brilliant colors are approximately +80 or -80. A circle drawn around the neutral point (a = b = 0) represents a color circle of constant chroma, where the angle h in degrees starting at red is a measure of the hue.

The equations for calculation of the CIELab color coordinates from the tristimulus values X, Y and Z is seen below.

$$L* = 116\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - 16 \qquad \text{Eq. (2.1)}$$

$$a* = 500\left[\left(\frac{X}{X_n}\right)^{\frac{1}{3}} - \left(\frac{Y}{Y_n}\right)^{\frac{1}{3}}\right] \qquad \text{Eq. (2.2)}$$

$$b* = 200\left[\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - \left(\frac{Z}{Z_n}\right)^{\frac{1}{3}}\right] \qquad \text{Eq. (2.3)}$$

$$C* = \sqrt{a^{*2} + b^{*2}} \qquad \text{Eq. (2.4)}$$

$$h = Arctan(\frac{b*}{a*}) \qquad \text{Eq. (2.5)}$$

The derivation of angular quadrant of h is as usual from the signs of a* and b*. The formulas here are only applied when each of the quotients $\frac{X}{X_n}$, $\frac{Y}{Y_n}$ and $\frac{Z}{Z_n}$ is greater than 0.008856, which is almost always the case with yarn, skein, woven or knitted substrates. If a quotient is smaller than 0.008856, the expression

$$7.787\ quotient + \frac{16}{116} \qquad \text{Eq. (2.6)}$$

Should be used instead of

$$(quotient)^{1/3} \qquad \text{Eq. (2.7)}$$

2.3 Recipe Preparation

Matching of a color sample demands extensive experience by the colorist concern the effects of individual dyestuffs on a combination of dyes. He/she must have a feeling for the relationship between the dye concentration employed and the appearance of the resulting dyeing. This relationship may also be determined by calculation. Computer color matching, on the basis of color measurement, enables quantitative determination of dyeing formulas for a sample to be matched. Formulation corrections may be similarly calculated.

Since around 1970, computer color matching has become increasingly common in the textile industry, and is meanwhile is most familiar application of color measurement. To a considerable extent this is due to instrument manufacturers, who presently offer matching systems high in performance and simple to operate. Such a matching system consists of a color measuring instrument, a computer with a permanent storage medium (e. g. magnetic disc), and the accompanying software for color measurement and formulation calculation.

The procedure involved in computer color matching compared to that in visual color matching will first briefly considered. The individual steps required in each procedure are compared in Table 2.1.

Table 2.1 Computer color matching versus visual color matching

Visual Color Matching	*Computer Color Matching*
One-time setup operation	
Coloristic experience	*Preparation of calibration dyeings of individual dyestuffs; measurement of the calibration dyeings, and storing of the dyestuff data*
Matching of a sample	
Viewing of the sample	*Measurement of the sample*
Selection of a dyestuff combination	*Selection of a series of dyestuffs*
Formula estimation with coloristic experience and a sample collection	*Entry of data and calculation of alternative formulas; selection of a formulation*
Dyeing of the formula	*Dyeing of the formula*
Visual comparison	*Measurement of the dyeing*
Estimation of the correction	*Calculation of the correction*
Selection of a new dyestuff combination if necessary	

Since the computer - in contrast to the dyer - has no coloristic experience, it must be "fed" with the appropriate information regarding the coloristic properties of dyes. Thus, calibration dyeings must be prepared of the individual dyes and measured in a one-time setup operation. These reflectance values, or the dyestuff data calculated from them, are then stored and can be used in computer color matching for years.

In visual matching of a color sample, the colorist first views the sample. This step corresponds to the measurement of the reflectance curve of the sample in computer color matching. As in visual matching, an experienced colorist is required for dye selection in computer color matching. After all, only dyes which are appropriate for the specific conditions (dyeing process. material. use of the dyed material. fastness. etc.) can be used. However, a relatively large number of dyes rather than only a single dye combination is generally specified for computer color matching.

While the colorist normally only estimates a single formulation, the computer calculates all formulations possible with the selected dyes. A variety of these is then

printed out together with costs and indices of metamerism (color deviation under artificial illumination). From these, the colorist selects an optimal formula for the specific application. A dyeing is then prepared -generally initially in the laboratory - with the selected formulation. As in visual matching, corrections are also necessary in computer matching, but usually require fewer steps. The computer program can similarly be used for determination of a corrected formula based on measurement of the match.

In visual color matching, it is not rare that although the initially selected dye combination results in a good match in daylight, the color deviates significantly from that of the standard under artificial illumination (incandescent lamp, fluorescent lamp). A new combination of dyes must be tried in this case. In contrast, in computer color matching the calculated indices of metamerism already indicate which dye combinations are suitable.

As shown by comparison with conventional matching, computer color matching cannot, and is not intended to replace the experienced colorist. Instead, the objective is to provide him/her with a technique which allows faster and more reliable matching, leaving more time for other responsibilities. Since a formulation which is optimal for the specific situation can be selected even prior to dyeing a sample, a technically better and/or more economical match will often be obtained as well. (Duran, 2001)

2.4 The Principles of Recipe Calculation

In recipe calculation the dye-stuff and their concentrations must be so properly selected that the reflectance value of the resultant color is as much as closer to the sample color.

The relationship between reflection (R) and concentration (C) must be known. (Duran, 1997)

$$C = f(R)\ ,\ R = f(C)$$ Eq. (2.8)

2.4.1 Kubelka Munk Formula

It is defined as;

$$\frac{K}{S} = \frac{(1-R)^2}{2\times R}$$ Eq. (2.9)

In a wavelength, the relationship between a reflection (R), absorption (K) and scattering (S) value is given. Generally; in colored textile materials, (K) is determined by dyestuff and (S) is determined by the textile material. (Duran, 1997)

2.4.2 Beer's Law

It is defined as;

$$A \times C = \frac{K}{S}$$ Eq. (2.10)

In this equation; the ratio of absorption (K) to scattering (S) gives concentration value. However; in this equation C must be multiplied by a coefficient A, which is different for every dyestuff.

The value of K/S which is calculated by the Kubelka-Munk formula for a definite wavelength is a measure of absorption of that colored textile material in this wavelength. The absorption of a colored textile material is equal to the sum of absorption of the textile material (Kt) and the absorption of the dyestuff (Kf). (Duran, 1997)

Then;

$$\frac{K}{S} = \frac{Kt+Kf}{S} = \frac{(1-R)^2}{2\times R}$$ Eq. (2.11)

2.4.3 Recipe Calculation

In order to calculate the recipe at least these values must be available:

- The remission value of the sample
- Which dyestuff used in the recipe calculation
- Which textile material will be dyed

In general; the values about dyestuff and measurement are saved to a computer with code numbers appropriate to the aim. In addition to this; the data about textile materials is needed.

The dyestuff which can be used for the recipe calculation and that are the plant has are saved to the computer by grouping with several criterions. The examples of these criterions are; specialty, the textile materials that they can dye, price, nuance etc. Until now; a grouping program that can make the optimum grouping with the all criterions we can think could not be developed. Still, the coloristic experience plays an important role in the choosing and grouping the dyestuff. On a large scale, the success of the recipe depends on this coloristic experience. (Duran, 1997)

The aim of this project is estimation of the recipe totally by developing an artificial intelligent model; not by the coloristic experience.

In this thesis; the experiments were done with the textile of 30/1 hosiery supreme. The experiments were done in the machine of TESA ELIAR TBB 100 automatic programmed laboratory type dyeing machine. In the dyeings, the dyeings of Procion Yellow H-E4R, Procion Crimson H-EXL and Procion Navy H-EXL were used from the group of Procion.

The chemical materials used in these experiments are sequestering agent (Dekol Sad), acetic acid (CH_3COOH), sodium carbonate (Na_2CO_3), sodium chloride (NaCl).

Table 2.2 Dyeing recipe

Procion Yellow H-E4R	0,05%
Procion Crimson H-EXL	0,00%
Procion Navy H-EXL	0,00%
pH 5.5-6 (Acetic acid)	2,5ml
Sequestering agent (Dekol Sad)	0,8ml
Sodium carbonate	4ml
Salt (NaCl)	2,6ml
Flotte ratio	1:8

Dyeing recipe is as shown in Table 2.2. The color measurements of the dyed textiles were done in the Data Colour (SF 600X) color measurement machine. Dyeing graph of the experiments done for this thesis is shown in Figure 2.4.

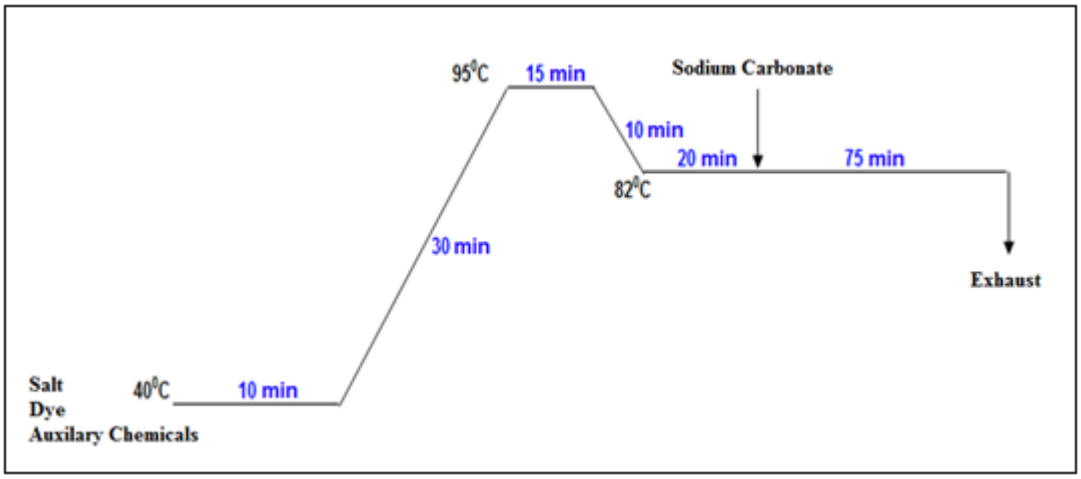

Figure 2.4 Dyeing graph of the experiments done for the thesis

CHAPTER THREE

ARTIFICIAL NEURAL NETWORKS

3.1 Introduction to Artificial Neural Networks

An artificial neural network can be defined as a system based on the operation of biological neural networks. In other words; it is an emulation of biological neural system. Although, nowadays computing is in a truly advanced level, there are certain tasks that a program made for a common microprocessor is unable to perform. Then; it can be said that artificial neural networks can be used for applications that are not able to be programmed with classical approaches.

Artificial Neural Networks analysis the examples of a certain event, by these examples it makes some generalization of that event. By this way, it makes some decisions related to that event.

As it can be happen for every application; artificial neural networks have some advantages and disadvantages (Haykin S., 1999).

3.1.1 Advantages of Artificial Neural Networks

- A neural network can perform tasks that a linear program can't perform.
- With the help of the parallel nature of the network; when an element of the neural network fails, it can continue without any problem
- A neural network learns and does not need to be reprogrammed.
- It can be implemented in any application.
- It can be implemented without any problem.

3.1.2 Disadvantages of Artificial Neural Networks

- In order to operate, the neural network needs training.
- The architecture of a neural network is different from the architecture of a microprocessor. Therefore, it is needed to be emulated.
- For large neural networks it requires high processing time.

3.2 The Biological Model

The human brain consists of a large number i.e. more than a billion of neural cells that process information. Each cell works like a simple processor and only the massive interaction between all cells and their parallel processing makes the brain's abilities possible

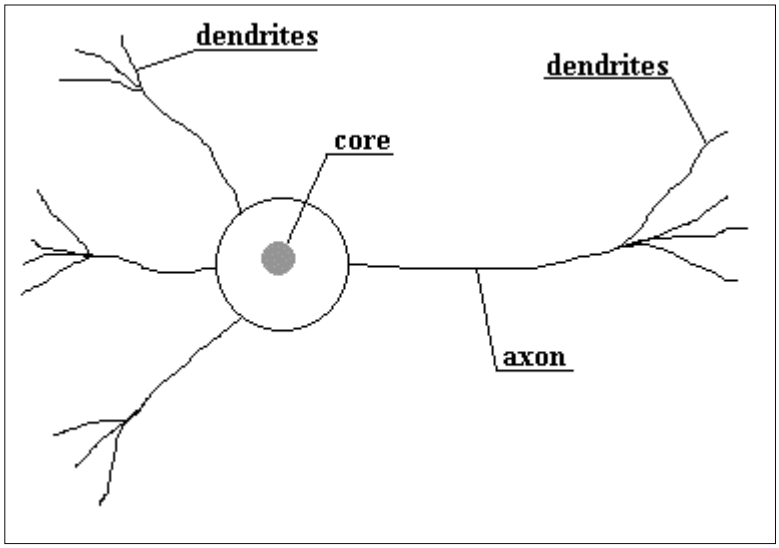

Figure 3.1 Structure of a neuron cell in the human brain

As shown in Figure 3.1, a neuron consists of a core, dendrites for incoming information and an axon with dendrites for outgoing information that is passed to connected neurons. Information is transported between neurons in form of electrical stimulations along the dendrites

Incoming information that reaches the neuron's dendrites is added up and then delivered along the neuron's axon to the dendrites at its end, where the information is

passed to other neurons if the stimulation has exceeded a certain threshold. In this case, the neuron is said to be activated.

If the incoming stimulation had been too low, the information will not be transported any further. In this case, the neuron is said to be inhibited.

The connections between the neurons are adaptive, what means that the connection structure is changing dynamically. It is commonly acknowledged that the learning ability of the human brain is based on this adaptation.

3.3 The Mathematical Model

When creating a mathematical model of a neural network, there are three basic components that must be taken into consideration. The first one is synapses and they are modeled as weights. The strength of the connection between an input and a neuron is measured by the value of the weight. An adder sums all the inputs modified by their respective weights and this process is called as linear combination. Finally, an activation function controls the amplitude of the output of the neuron. An acceptable range of output is usually between 0 and 1, or -1 and 1. Mathematically, his process is defined in Figure 3.2

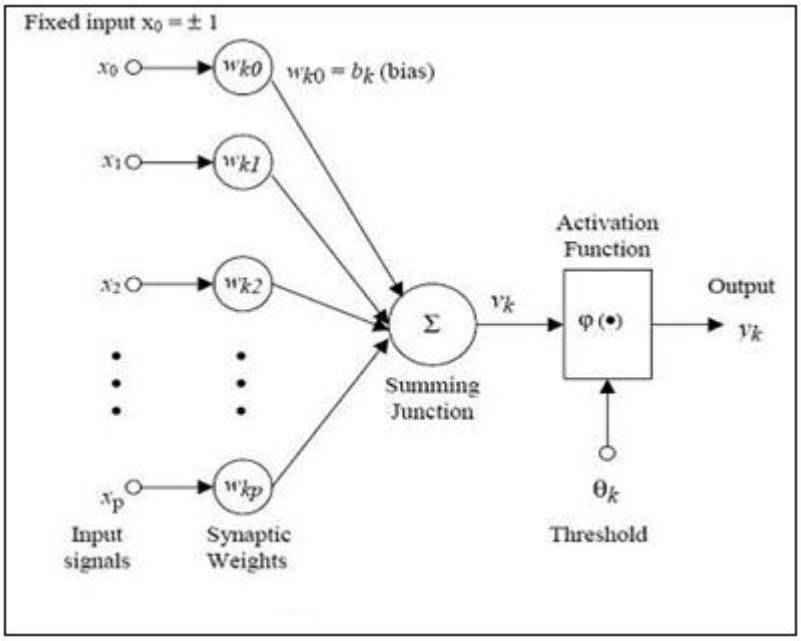

Figure 3.2 Mathematical model of a neural network

From this model the interval activity of the neuron can be shown to be:

$$v_k = \sum_{j=1}^{p} w_{kj}\, x_j \qquad \text{Eq. (3.1)}$$

The output of the neuron, yk, would therefore be the outcome of some activation function on the value of vk.

3.3.1 Activation Functions

The activation function acts as a squashing function. That is; it acts such that the output of a neuron in a neural network is between certain values (usually 0 and 1, or -1 and 1). In general there are three types of activation functions and it is denoted by ϕ(.)

3.3.1.1 Threshold Function

The Threshold Function takes a value of 0 if the summed input is less than a certain threshold value (v), and the value 1 if the summed input is greater than or equal to the threshold value.

$$\varphi(\mathrm{v}) = \begin{cases} 1 \text{ if } \mathrm{v} \geq 0 \\ 0 \text{ if } \mathrm{v} < 0 \end{cases} \qquad \text{Eq. (3.2)}$$

3.3.1.2 Piecewise-Linear Function

Piecewise-Linear Function again can take the values of 0 or 1, but it can also take values between that depending on the amplification factor in a certain region of linear operation.

$$\varphi(\mathrm{v}) = \begin{cases} 1 \text{ if } \mathrm{v} \geq \frac{1}{2} \\ \mathrm{v} \text{ if } -\frac{1}{2} > v > \frac{1}{2} \\ 0 \text{ if } \mathrm{v} \leq -\frac{1}{2} \end{cases} \qquad \text{Eq. (3.3)}$$

3.3.1.3 Sigmoid Function

The Sigmoid Function can range between 0 and 1, but it is also sometimes useful to use the -1 to 1 range. An example of the sigmoid function is the hyperbolic tangent function:

$$\varphi(v) = \tanh\left(\frac{v}{2}\right) = \frac{1-\exp(-v)}{1+\exp(-v)} \qquad \text{Eq. (3.4)}$$

The common nonlinear activation functions are shown in Figure 3.3

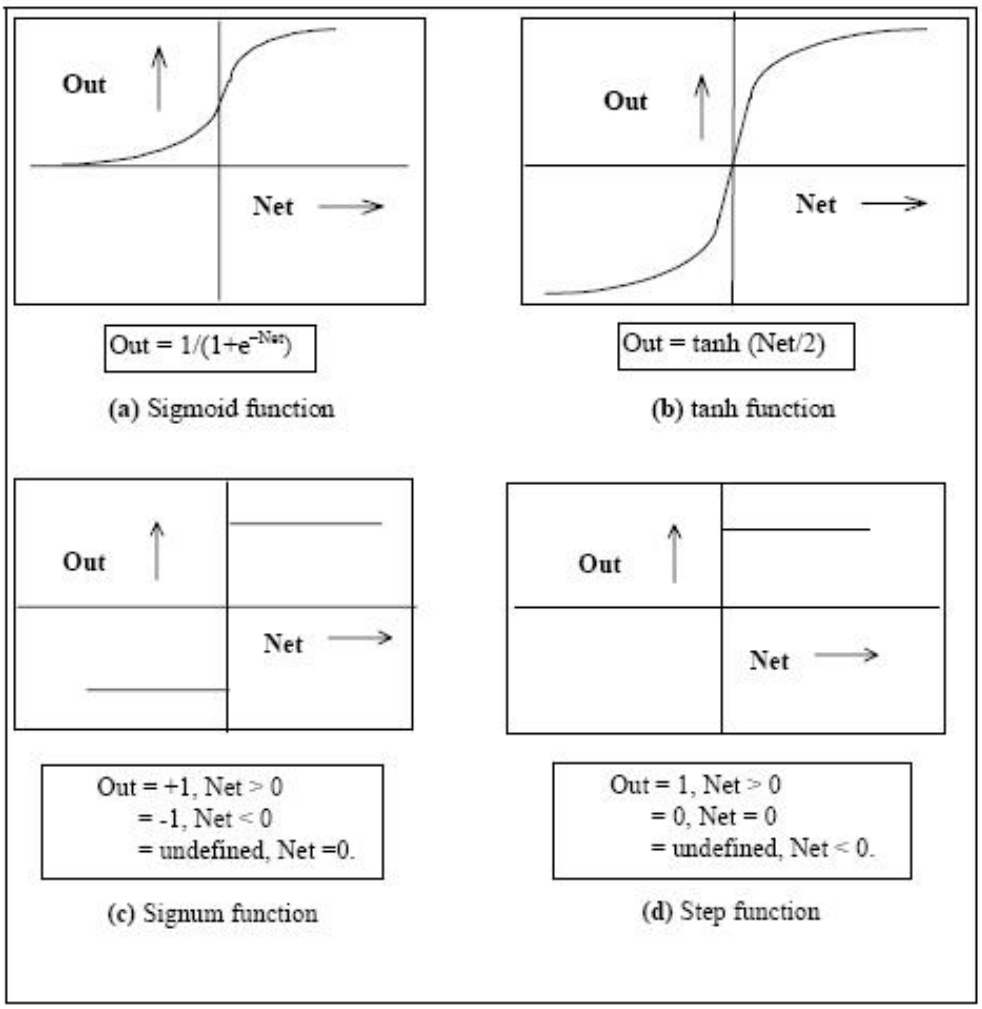

Figure 3.3 Common non-linear functions used for synaptic inhibition. Soft non-linearity: a)Sigmoid and b)Tanh; Hard non-linearity: c)Sigmoid and d)Step

3.4 Neural Network Paradigms

In order to train a neural network, three learning paradigms can be used. These are supervised learning, unsupervised learning and reinforcement learning

3.4.1 Supervised Learning

The task of the supervised learner is to predict the value of the function for any valid input object after having seen a number of training examples (i.e. pairs of input and target output). To achieve this, the learner has to generalize from the presented data to unseen situations in a "reasonable" way

In supervised learning technique, the input and the expected output of the system are provided. Artificial Neural Network is used to model the relationship between the input and the output. Given an input set *x*, and a corresponding output set *y*, an optimal rule is to be determined such that:

$$y = f(x) + e \qquad \text{Eq. (3.5)}$$

In this equation, *e* is an approximation error and it is needed to be minimized. With the input values provided to the network, it produces a result. This result is compared to the desired result, and the error signal e is used to update the network weight vectors. When we want the network to reproduce the characteristics of a certain relationship, supervised learning is useful.

Approximately 70 or 80 % of real world applications use supervised learning. The major applications areas of supervised learning are; bioinformatics, cheminformatics, handwriting recognition, information retrieval, object recognition in computer vision, optical character recognition, spam detection, pattern recognition, speech recognition, forecasting fraudulent financial statements.

3.4.2 Unsupervised Learning

In unsupervised learning, the data and a cost function which is a function of the system input and output are used. The ANN is trained in order to minimize the cost function by finding a suitable input-output relationship.

The aim is to minimize the cost function through a proper selection of f (the relationship between x, and y) for a given input set x, and a cost function $g(x, y)$ of the input and output sets. In all iterations, the trainer provides the input to the network, and the network produces a result. This result is put into the cost function, and the total cost is used to update the weights. Weights are continuously updated until the system output produces a minimal cost. Unsupervised learning is useful in situations where a cost function is known, but a data set is not known that minimizes that cost function over a particular input space.

One of the application examples of unsupervised learning is clustering. Another example is blind source separation based on Independent Component Analysis (ICA).

The most commonly used unsupervised learning algorithms are the self-organizing map (SOM) and adaptive resonance theory (ART). The SOM is a topographic organization. In this organization nearby locations in the map represent inputs with similar properties. The ART model allows the number of clusters to vary with problem size and by the way, it lets the user control the degree of similarity between members of the same clusters by means of a user-defined constant called the vigilance parameter. ART networks are also used for many pattern recognition tasks, such as automatic target recognition and seismic signal processing.

3.4.3 Reinforcement Learning

In reinforcement learning, data x is usually not given. However it is generated by an agent's interactions with the environment. At each point in time t, the agent performs an action y_t and the environment generates an observation x_t and an instantaneous cost c_t, according to some dynamics. These dynamics are usually

unknown. The aim is to discover a *policy* for selecting actions that minimizes some measure of a long-term cost, i.e. the expected cumulative cost. The environment's dynamics and the long-term cost for each policy are usually unknown, but can be estimated.

ANNs are frequently used in reinforcement learning as part of the overall algorithm.

The tasks that we can use reinforcement learning are control problems, games and other sequential decision making tasks.

3.5 Types of Neural Networks

3.5.1 Feedforward Neural Network

The feedforward neural network was the first and one of the simplest types of artificial neural network designed. In this network, the information moves in only forward direction, from the input nodes, through the hidden nodes and to the output nodes. There are no cycles or loops in the network. The structure of feedforward neural network is shown in Figure 3.4

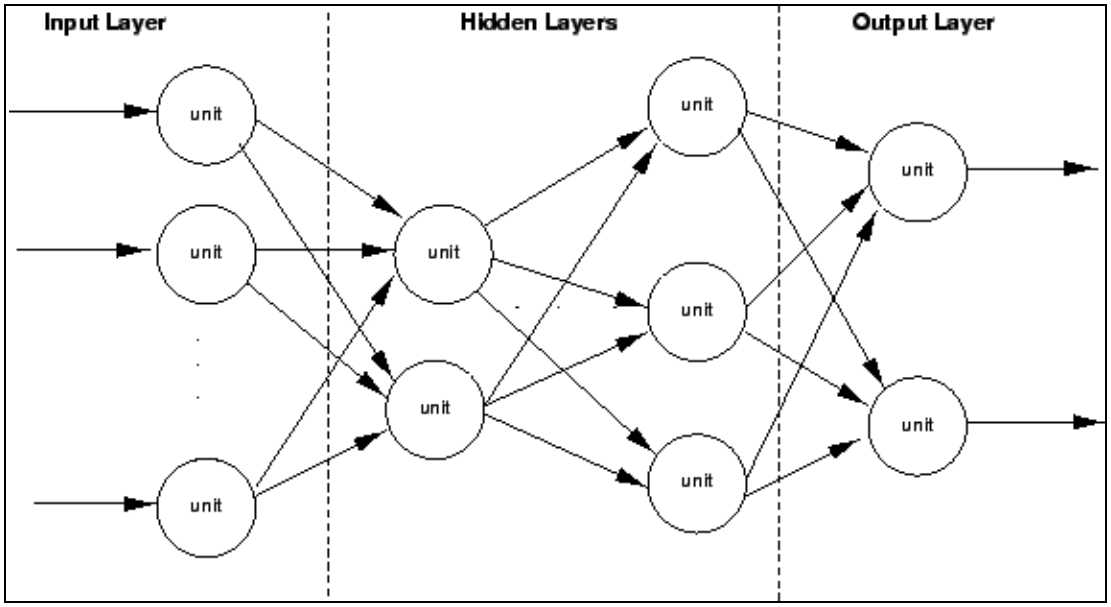

Figure 3.4 Feedforward neural network

3.5.1.1 Single Layer Perceptron

Single-layer perceptron network is the earliest kind of neural network. It consists of a single layer of output nodes; the inputs are fed directly to the outputs via a series of weights as shown in Figure 3.5. By this way, it can be considered as the simplest kind of feed-forward network. In each node; the sum of the products of the weights and the inputs is calculated, and if the value is above some threshold, the neuron fires and takes the activated value. Generally threshold is 0, activated value is 1 and deactivated value is -1. Neurons with this kind of activation function are also called artificial neurons or linear threshold units. In the literature the term perceptron often refers to networks consisting of just one of these units. A similar neuron was described by Warren McCulloch and Walter Pitts in the 1940s.

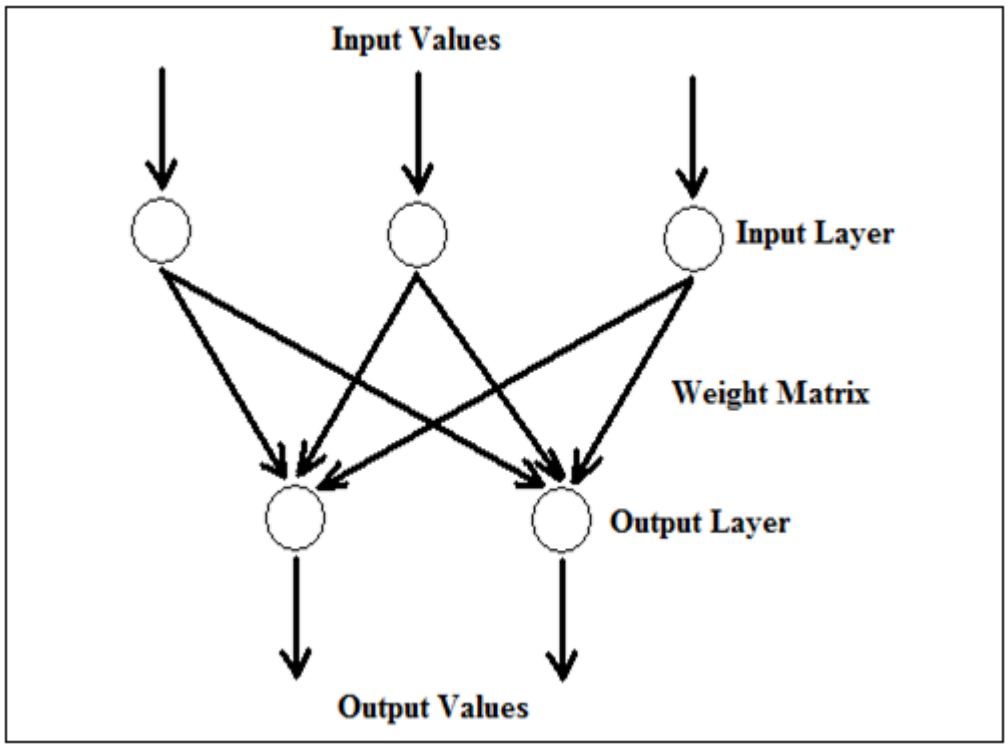

Figure 3.5 Single Layer Perceptron

A perceptron can be created using any values for the activated and deactivated states as long as the threshold value lies between the two. In most perceptrons outputs are 1 or -1 with a threshold of 0 and it is known that such networks can be

trained more quickly than networks created from nodes with different activation and deactivation values.

Perceptrons can be trained by a simple learning algorithm called the delta rule. It calculates the errors between calculated output and sample output data. Then it uses this to adjust the weights. This is an implementation of gradient descent.

Single-layer perceptrons are only able to learn linearly separable patterns. Although a single threshold unit is quite limited in its computational power, networks of parallel threshold units can approximate any continuous function from a compact interval of the real numbers into the interval.

A single-layer neural network can compute a continuous output instead of a step function. A common choice is the so-called sigmoid function

$$y = \frac{1}{1+e^{-x}} \qquad \text{Eq. (3.6)}$$

3.5.1.2 Multi Layer Perceptron

This class of networks consists of multiple layers of computational units which are usually interconnected in a feed-forward way as shown in Figure 3.6. Each neuron in one layer has directly connected to the neurons of the subsequent layer. In many applications, as an activation function, the units of these networks apply a sigmoid function.

The universal approximation theorem for neural networks states that every continuous function that maps intervals of real numbers to some output interval of real numbers can be approximated arbitrarily closely by a multi-layer perceptron with just one hidden layer. This result holds only for restricted classes of activation functions, e.g. for the sigmoidal functions.

In multi-layer neural networks, a variety of learning techniques are used. However, the most popular one is back-propagation. In back-propagation, the output values are compared with the correct answer in order to compute the value of some

predefined error-function. Then, the error is fed back through the network by using various techniques. The algorithm adjusts the weights of each connection in order to reduce the value of the error function by some small amount with the help of this information. After repeating this process for a sufficiently large number of training cycles, the network will usually converge to some state where the calculated error is small. In this case, it can be said that the network has learned a certain target function. To adjust weights properly, a general method for non-linear optimization that is called gradient descent can be applied. For applying gradient descent, the derivative of the error function with respect to the network weights is calculated, and the weights are then changed such that the error decreases. Therefore, back-propagation can only be applied on networks with differentiable activation functions.

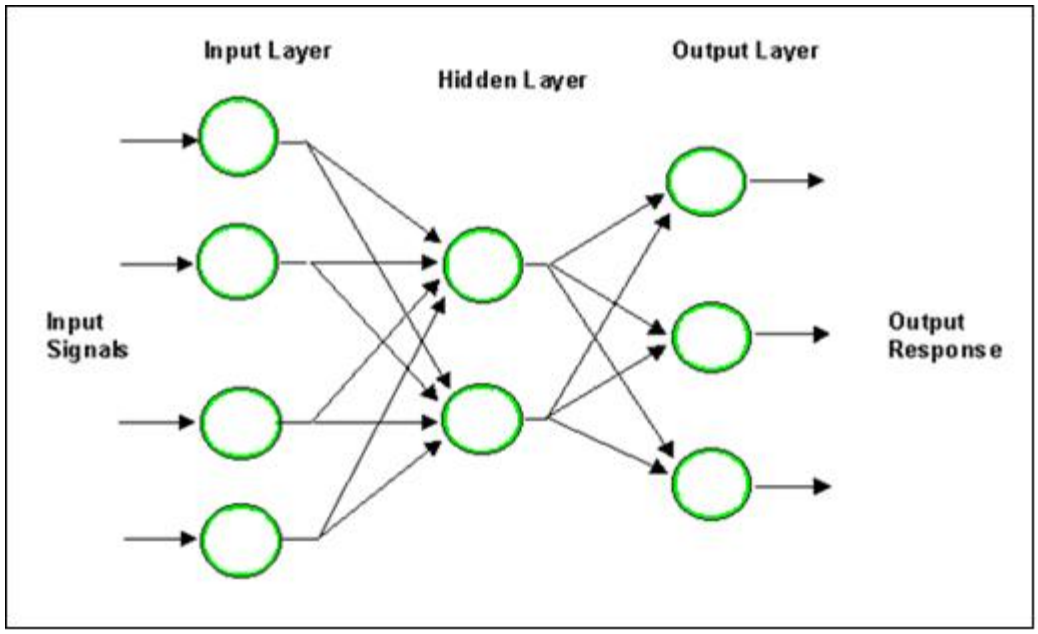

Figure 3.6 Multilayer perceptron

Generally, teaching a network to perform well, even on samples that were not used as training samples is a very complicated issue that it requires additional techniques. This is especially important for cases where only very limited numbers of training samples are available. In this system, there is a risk of overfitting of the network with the training data and therefore it fails to capture the true statistical process generating the data. Computational learning theory is concerned with training classifiers on a limited amount of data. In the context of neural networks a

simple technique called early stopping, often ensures that the network will generalize well to examples not in the training set.

Other most common problems of the back-propagation algorithm are the speed of convergence and the possibility of ending up in a local minimum of the error function. However, today there are practical solutions that make back-propagation in multi-layer perceptron the solution for many machine learning tasks.

3.5.2 Radial Basis Function (RBF) Network

Radial Basis Function Networks are a type of ANN where the hidden layer is composed of radial-basis curves. The structure of RBF is shown in Figure 3.7. RBFs can be used to perform interpolation in multidimensional space.

RBF is a function which has built into a distance criterion with respect to a center. Radial basis functions have been applied in the area of neural networks where they may be used as a replacement for the sigmoidal hidden layer transfer characteristic in MLP. RBF networks have two layers of processing: In the first layer, input is mapped onto each RBF in the 'hidden' layer. Usually Gaussian is used as the RBF. In regression problems the output layer is then a linear combination of hidden layer values which represent mean predicted output. In classification problems the output layer is generally a sigmoid function of a linear combination of hidden layer values which are representing a posterior probability.

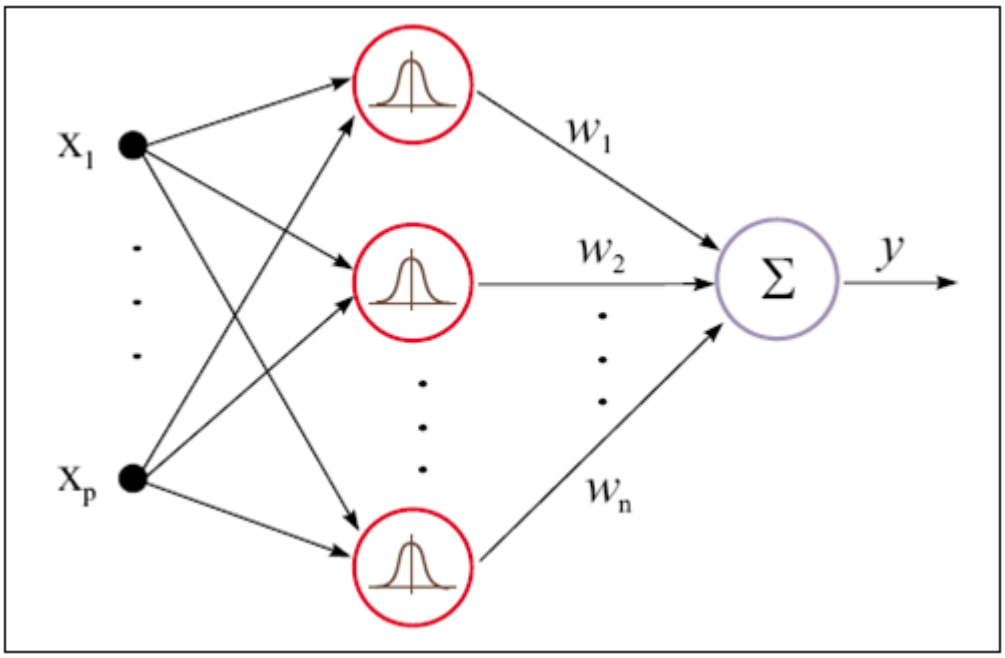

Figure 3.7 Radial Basis Function (RBF) Neural Network

Unlike the MLP, RBF networks have the advantage of not suffering from local minima because, the only parameters that are adjusted in the learning process are the linear mapping from hidden layer to output layer. Linearity provides that the error surface is quadratic and therefore has a single minimum which is easily found.

RBF networks have the disadvantage of requiring good coverage of the input space by radial basis functions. RBF centers are determined with reference to the distribution of the input data, but without reference to the prediction task. As a result, representational resources may be wasted on areas of the input space that are irrelevant to the learning task. A common solution is to associate each data point with its own centre, although this can make the linear system to be solved in the final layer rather large, and requires shrinkage techniques to avoid overfitting.

3.5.3 Kohonen Self-Organizing Network

Kohonon's SOM is a kind of unsupervised learning. The aim of SOM is to discover some underlying structure of the data. However, the kind of structure that will be discovered is very different than PCA or vector quantization.

Kohonen's SOM is called a topology-preserving map because there is a topological structure imposed on the nodes in the network. A topological map is simply a mapping that preserves neighborhood relations.

In the networks explained so far, the geometric arrangements of output nodes have been ignored. Each node in a given layer was connected with all of the nodes in the upper and/or lower layer. In SOM physical arrangement of these nodes is taken into consideration. Nodes that are "close" together are going to interact differently than nodes that are "far" apart.The structure of Kohonen self organization network is shown in Figure 3.8.

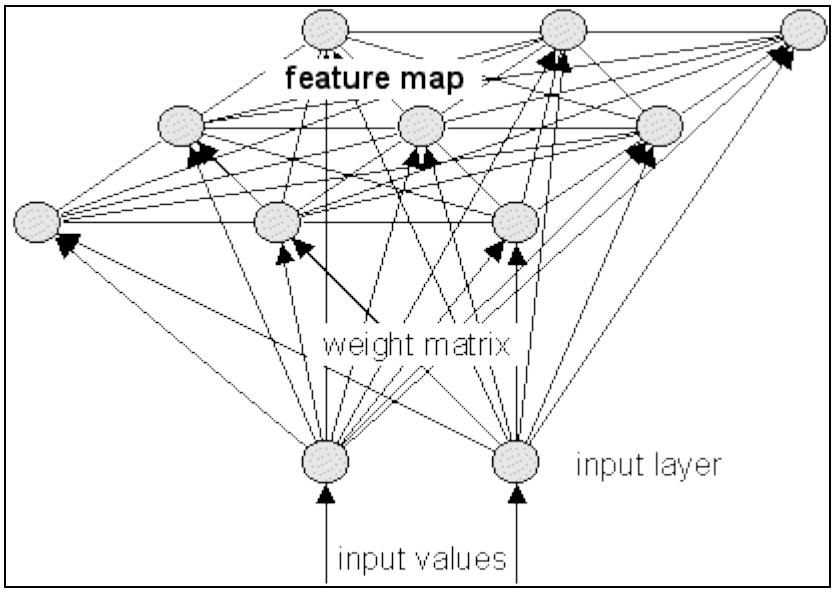

Figure 3.8 Kohonen Self-Organization Network

Neurons tend to cluster in groups in the brain. The connections within the group are much greater than the connections with the neurons outside of the group. Kohonen's network tries to imitate this behavior of the brain this.

3.5.4 Recurrent Network

A recurrent network is defined as a network in which either the network's hidden unit activations or output values are fed back into the network as inputs. In other words; a recurrent neural network (RNN) is a class of neural network where connections between units form a directed cycle. This creates an internal state of the

network which allows it to exhibit dynamic temporal behavior. The structure of recurrent neural networks is shown in Figure 3.9.

Both when analyzing their behavior and training them, recurrent neural networks must be approached differently from feedforward neural networks. Recurrent neural networks can also behave chaotically. Usually, in order to model and analyze the recurrent neural networks, dynamical systems theory is used. While a feedforward network propagates data linearly from input to output, recurrent networks (RN) also propagate data from later processing stages to earlier stages.

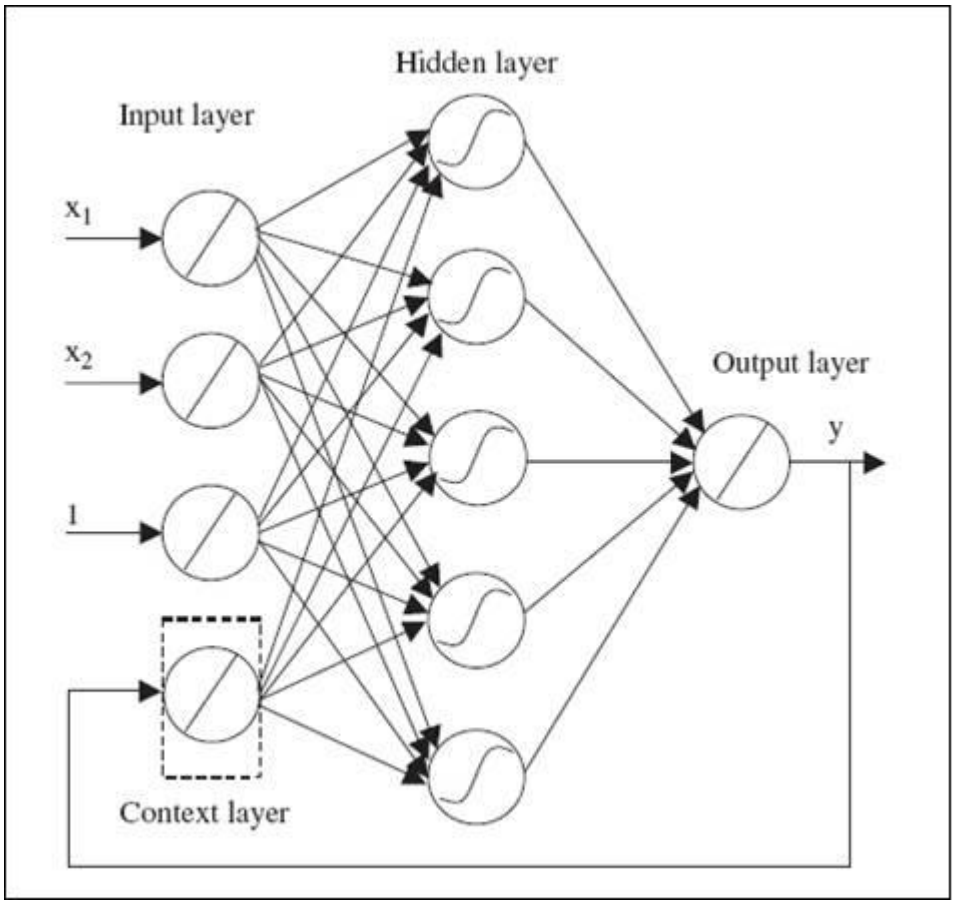

Figure 3.9 Recurrent Neural Networks

3.6 Applications of Neural Networks

The utility of artificial neural network models lies in the fact that they can be used to infer a function from observations. This is particularly useful in applications where

the complexity of the data or task makes the design of such a function by hand impractical.

The tasks to which artificial neural networks are applied tend to fall within the following broad categories: Function approximation, or regression analysis, including time series prediction, fitness approximation and modeling, classification, including pattern and sequence recognition, novelty detection and sequential decision making, data processing, including filtering, clustering, blind source separation and compression, robotics, including directing manipulators, Computer numerical control.

Application areas include system identification and control (vehicle control, process control), game-playing and decision making (backgammon, chess, racing), pattern recognition (radar systems, face identification, object recognition and more), sequence recognition (gesture, speech, handwritten text recognition), medical diagnosis, financial applications (automated trading systems), data mining (or knowledge discovery in databases, "KDD"), visualization and e-mail spam filtering.

3.6.1 Textile Applications of Artificial Neural Networks and Fuzzy Logic

As mentioned in the previous part, the use of artificial neural networks (ANN) is becoming popular during last decades. ANN has been trained to perform complex functions in various fields including; control, robotics, pattern recognition, forecasting, medicine, power systems, manufacturing. It is also possible to find many applications in textile field. Many textile researchers have used ANN for modeling, predicting, and determining the properties and behavior of the fibers, yarns, and fabrics. In addition, ANN has also proved useful for many prediction-related problems in textiles such as the prediction of characteristics of textiles; identification, classification and analysis of defects; process optimization; and marketing and planning. For example, ANN have been used for predicting thermal resistance of textile fabrics, properties of worsted fabrics, ring yarn elongation from cotton fiber properties, and for predicting spirality of fully relaxed single jersey fabrics. (Ergan Z. H., Çukul D., Öztürk M. M. 2009)

There are some studies based on artificial neural networks in textile industry. Some of these studies are as below.

The study of M. Senthilkumar and N. Selvakumar named as achieving expected depth of shade in reactive dye application using artificial neural network technique. In this study the correct duration of dyeing is aimed to find. In order to find this duration artificial neural networks was used as a method. In this application, the trained neural network has an average error of 1% with respect to the dyeing time. Hence the neural network developed can be used to determine the primary exhaustion time and fixation time for producing the expected depth of shade with high exhaustion reactive dye on cotton fabric. (Senthilkumar M., Selvakumar N., 2005)

The study of M. Senthilkumar named as modeling of CIELab values in vinyl sulphone dye application using feed-forward neural networks. This article is concerned with the CIELab values prediction based on a neural network developed for cotton fabric dyed with vinyl sulphone reactive dye. The neural network used here is a multilayer neural network. The results obtained from the network gives an average error of around 2% for vinyl sulphone dyes used for training the network in prediction the Lab values. (Senthilkumar M., 2006)

The study of Golob D, Zupan J., Osterman D. P. named as the use of artificial neural networks for color prediction in textile printing. This article is concerned with the usage of neural networks for prediction of dyes in textile printing paste preparation. In this study 1340 samples were used either a single dye or a combination of two dyes. In this application, L, a, b values were used. The success percentage of both dyes correct is 96.8% and only one dye correct is 3.2% and both dyes incorrect is 0%. Thus, the system predicts at least one dye of two. (Golob D., Zupan J., Osterman D.P., n.d)

The study of Çeven E. K., Özdemir Ö. named as using fuzzy logic to evaluate and predict chenille yarn’s shrinkage behavior. In this study; a fuzzy logic system is used

to determine the effects of yarn parameters on the boiling shrinkage behavior of chenille yarns. According to the results, chenille yarns with higher twist levels and shorter pile lengths have lower shrinkage values and the yarn count has a significant effect on shrinkage. The comparison of the results obtained from the fuzzy logic model and the experiments shows that there is a strong relationship between the measured and predicted yarn shrinkage values. (Çeven E. K., Özdemir Ö., 2007)

The study of Kandi S. G., Tehran M. A. named as color recipe prediction by genetic algorithm. This article is concerned with a new method of color recipe prediction is proposed by applying genetic algorithm. As mentioned in this article this method is able to do both spectrophotometric and colorimetric color matching based on its fitness function. In this application, it is shown that the problem of the limitation of colorant numbers in colorimetric and color matching can be solved. In addition this algorithm is capable of decreasing the color difference under second illuminant and reduces metamerism problem by applying a fitness function based on the color differences under two illuminants. (Kandi S. G., Tehran M. A., 2006)

The study of Bhattacharjee D. and Kothari V. K. named as A Neural Network System for Prediction of Thermal Resistance of Textile Fabrics. The objective of this paper is to report a study on the predictability of the steady-state and transient thermal properties of fabrics using a feed-forward, back-propagation artificial neural network system. A comparison was made with two different network architectures, one with two sequential networks working in tandem fed with a common input and another with a single network that gave two outputs. A three-layered network was used in both the cases. The networks were then subjected to a set of untrained inputs and the output thermal properties, namely thermal resistance and Qmax, were compared with the values obtained experimentally. The architecture with two Networks working in tandem with a common set of inputs gave better. It was found that different networks for different outputs gave a better prediction of the thermal behavior of the fabrics. This study therefore, shows that artificial neural networks can be used as a tool to predict the steady-state and transient thermal behavior of a fabric satisfactorily. (Bhattacharjee D., Kothari V. K., 2007)

In this thesis color recipe prediction is done with neural networks and fuzzy logic. The difference of this project from the others above is, it is done with three different methods, which are MLP NN, RBF NN and anfis of fuzzy logic. With all of these methods, some training processes were done in MATLAB by changing various parameters, as a result the response of the system was observed. According to the results, RBF was the best method of the three because, RBF learns faster than the others and it provides more successful results. It has also the 0% error percentage with less training time and it is also more stable than the other training structures used.

CHAPTER FOUR

FUZZY LOGIC

4.1 Introduction to Fuzzy Logic

The concept of Fuzzy Logic (FL) was conceived by Lotfi Zadeh and presented not as a control methodology, but as a way of processing data by allowing partial set membership rather than crisp set membership or non-membership. This approach to set theory was not applied to control systems until the 70's due to insufficient small-computer capability prior to that time. Professor Zadeh reasoned that people do not require precise, numerical information input, and yet they are capable of highly adaptive control. If feedback controllers could be programmed to accept noisy, imprecise input, they would be much more effective and perhaps easier to implement.

FL was conceived as a better method for sorting and handling data but has proven to be an excellent choice for many control system applications since it mimics human control logic. It can be built into anything from small, hand-held products to large computerized process control systems. It uses an imprecise but very descriptive language to deal with input data more like a human operator. It is very robust and forgiving of operator and data input and often works when first implemented with little or no tuning.

4.2 Features of Fuzzy Logic

FL offers several unique features that make it a particularly good choice for many control problems.

- It is inherently robust since it does not require precise, noise-free inputs and can be programmed to fail safely if a feedback sensor quits or is destroyed. The output control is a smooth control function despite a wide range of input variations.

- Since the FL controller processes user-defined rules governing the target control system, it can be modified and tweaked easily to improve or drastically to alter system performance. New sensors can easily be incorporated into the system simply by generating appropriate governing rules.
- FL is not limited to a few feedback inputs and one or two control outputs, nor is it necessary to measure or compute rate-of-change parameters in order for it to be implemented. Any sensor data that provides some indication of a system's actions and reactions is sufficient. This allows the sensors to be inexpensive and imprecise thus keeping the overall system cost and complexity low.
- Because of the rule-based operation, any reasonable number of inputs can be processed (1-8 or more) and numerous outputs (1-4 or more) generated, although defining the rule-base quickly becomes complex if too many inputs and outputs are chosen for a single implementation since rules defining their interrelations must also be defined. It would be better to break the control system into smaller chunks and use several smaller FL controllers distributed on the system, each with more limited responsibilities.
- FL can control nonlinear systems that would be difficult or impossible to model mathematically. This opens doors for control systems that would normally be deemed unfeasible for automation.

4.3 Linguistic Variables

In 1973, Professor Lotfi Zadeh proposed the concept of linguistic or "fuzzy" variables. It must be thought of them as linguistic objects or words, rather than numbers. The sensor input is a noun, e.g. "temperature", "displacement", "velocity", "flow", "pressure", etc. Since error is just the difference, it can be thought of the same way. The fuzzy variables themselves are adjectives that modify the variable (e.g. "large positive" error, "small positive" error, "zero" error, "small negative" error, and "large negative" error). As a minimum, one could simply have "positive", "zero", and "negative" variables for each of the parameters. Additional ranges such as "very large" and "very small" could also be added to extend the responsiveness to exceptional or very nonlinear conditions, but aren't necessary in a basic system.

4.4 The Rule Matrix

In the part 4.3 the concept of linguistic variables was presented. The fuzzy parameters of error (command-feedback) and error-dot (rate-of-change-of-error) were modified by the adjectives "negative", "zero", and "positive". To picture this, imagine the simplest practical implementation, a 3-by-3 matrix. The columns represent "negative error", "zero error", and "positive error" inputs from left to right. The rows represent "negative", "zero", and "positive" "error-dot" input from top to bottom. This planar construct is called a rule matrix. It has two input conditions, "error" and "error-dot", and one output response conclusion (at the intersection of each row and column). In this case there are nine possible logical product output response conclusions.

Although not absolutely necessary, rule matrices usually have an odd number of rows and columns to accommodate a "zero" center row and column region. This may not be needed as long as the functions on either side of the center overlap somewhat and continuous dithering of the output is acceptable since the "zero" regions correspond to "no change" output responses the lack of this region will cause the system to continually hunt for "zero". It is also possible to have a different number of rows than columns. This occurs when numerous degrees of inputs are needed. The maximum number of possible rules is simply the product of the number of rows and columns, but definition of all of these rules may not be necessary since some input conditions may never occur in practical operation. The primary objective of this construct is to map out the universe of possible inputs while keeping the system sufficiently under control.

4.5 Membership Functions

The membership function is a graphical representation of the magnitude of participation of each input. It associates a weighting with each of the inputs that are processed, define functional overlap between inputs, and ultimately determines an output response. The rules use the input membership values as weighting factors to determine their influence on the fuzzy output sets of the final output conclusion. Once the functions are inferred, scaled, and combined, they are defuzzified into a

crisp output which drives the system. There are different membership functions associated with each input and output response. There are some features to note below.

SHAPE - triangular is common, but bell, trapezoidal, haversine and, exponential have been used. More complex functions are possible but require greater computing overhead to implement.

HEIGHT or magnitude (usually normalized to 1)

WIDTH (of the base of function)

SHOULDERING (locks height at maximum if an outer function. Shouldered functions evaluate as 1.0 past their center)

CENTER points (center of the member function shape)

OVERLAP (N&Z, Z&P, typically about 50% of width but can be less).

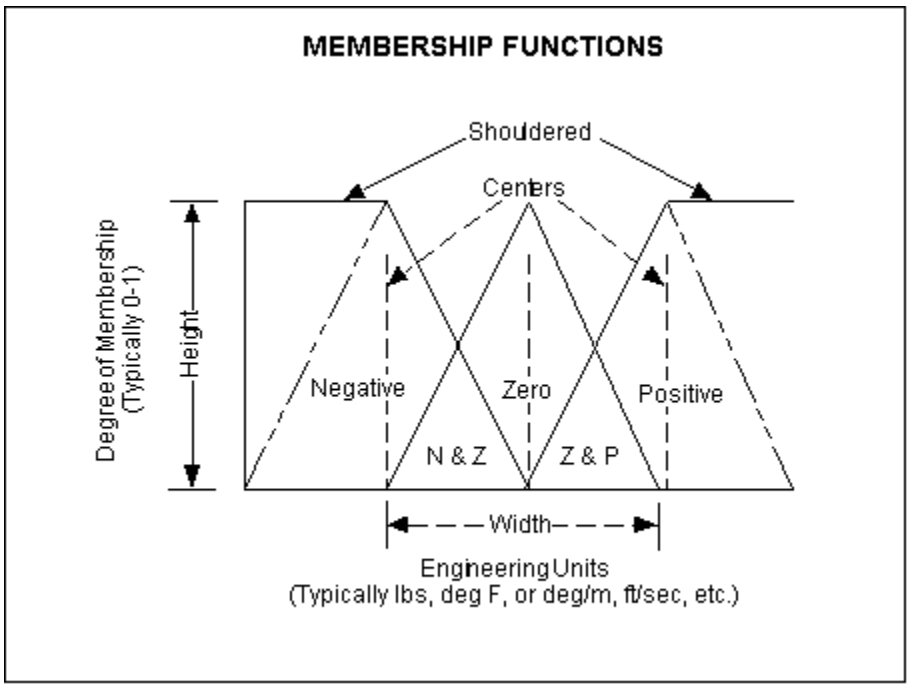

Figure 4.1 The features of a membership function

Figure 4.1 illustrates the features of the triangular membership function which is used in this example because of its mathematical simplicity. Other shapes can be used but the triangular shape lends itself to this illustration.

The degree of membership (DOM) is determined by plugging the selected input parameter (error or error-dot) into the horizontal axis and projecting vertically to the upper boundary of the membership function(s).

The degree of membership for an "error" of -1.0 projects up to the middle of the overlapping part of the "negative" and "zero" function so the result is "negative" membership = 0.5 and "zero" membership = 0.5. Only rules associated with "negative" & "zero" error will actually apply to the output response. This selects only the left and middle columns of the rule matrix.

For an "error-dot" of +2.5, a "zero" and "positive" membership of 0.5 is indicated. This selects the middle and bottom rows of the rule matrix. By overlaying the two regions of the rule matrix, it can be seen that only the rules in the 2-by-2 square in the lower left corner (rules 4, 5, 7, 8) of the rules matrix will generate non-zero output conclusions. The others have a zero weighting due to the logical AND in the rules.

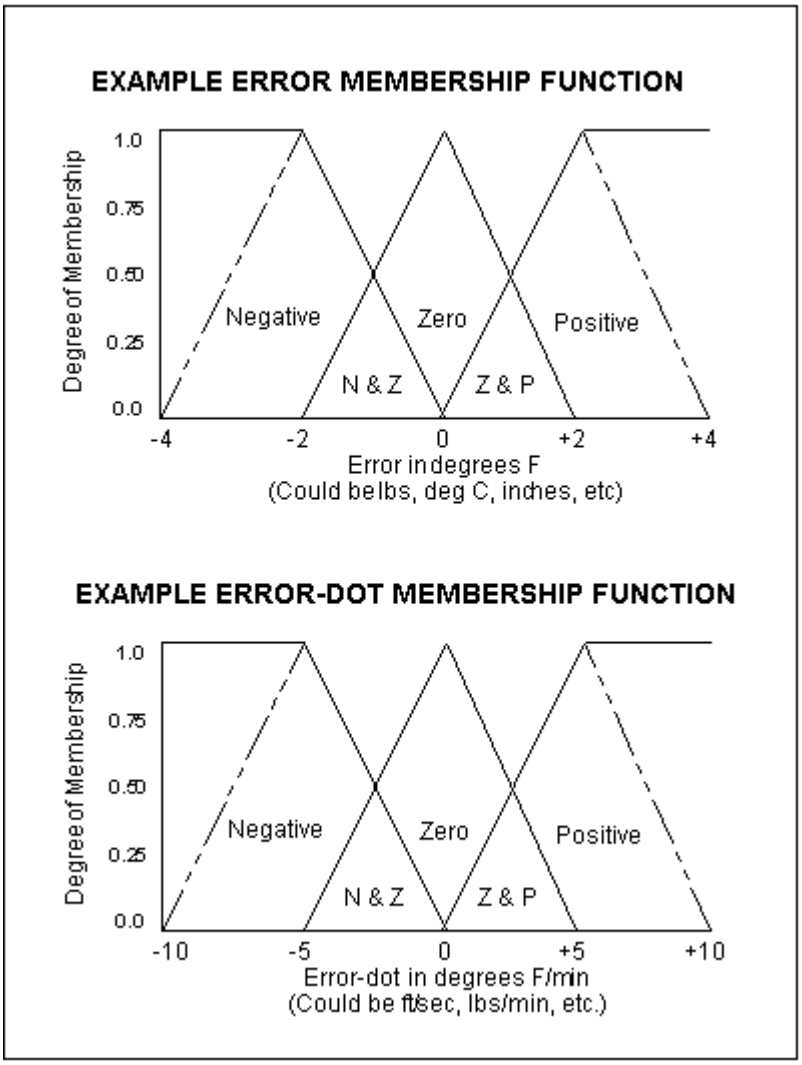

Figure 4.2 A sample case

In Figure 4.2 consider an "error" of -1.0 and an "error-dot" of +2.5. These particular input conditions indicate that the feedback has exceeded the command and is still increasing.

4.6 Application Areas

Fuzzy logic is used in a wide range of areas. These areas are air conditioners, automobile and other vehicle subsystems, such as automatic transmissions, abs and cruise control (e.g. tokyo monorail), cameras, digital image processing, such as edge detection, dishwashers, elevators, fuzzy logic has also been incorporated into some microcontrollers and microprocessors, for instance, the freescale 68hc12, hydrometeor classification algorithms for polarimetric weather radar, language filters

on message boards and chat rooms for filtering out offensive text, the massive engine used in the *lord of the rings* films, which helped huge scale armies create random, yet orderly movements, mineral deposit estimation, pattern recognition, in remote sensing, rice cookers, video game, artificial intelligence, washing machines and other home appliances

4.7 Anfis

ANFIS is Adaptive Neuro-Fuzzy training of Sugeno-type FIS. ANFIS uses a hybrid learning algorithm to identify the membership function parameters of single-output, Sugeno type fuzzy inference systems (FIS). A combination of least-squares and backpropagation gradient descent methods are used for training FIS membership function parameters to model a given set of input/output data.

[FIS, ERROR] = ANFIS(TRNDATA) tunes the FIS parameters using the input/output training data stored in TRNDATA. For an FIS with N inputs, TRNDATA is a matrix with N+1 columns where the first N columns contain data for each FIS input and the last column contains the output data. ERROR is the array of root mean square training errors (difference between the FIS output and the training data output) at each epoch. ANFIS uses genfis1 to create a default FIS that is used as the starting point for ANFIS training.

The training process stops whenever the designated epoch number is reached or the training error goal is achieved. STEPSIZE is an array of step sizes. The step size is increased or decreased by multiplying it by the step size increase or decrease rate as specified in the training options. Entering NaN for any option will select the default value.

CHAPTER FIVE

APPLICATION

5.1 General Information

In this thesis, we have examined color recipe prediction in textile industry. As it is mentioned previously, recipe prediction is done in various methods. Some of these methods are done with the human experience; some of them are performed with the help of computer programs.

In this project, the systems of Artificial Neural Network (ANN) and Fuzzy Logic are used in order to predict the recipe. In ANN, the methods of Multi Layer Perceptron (MLP) and Radial Basis Function (RBF) were used. In fuzzy logic, the method of Adaptive Neuro-Fuzzy Inference System (ANFIS) was used. All of the applications were done in MATLAB platform. In every application, the data obtained from the textile experiments. These data are generated in the forms of CIELab and Reflectance. These data are given in the table below.

These applications were done with various data sets and changing some parameters. Generally, all of the applications were done with a training set of 250 samples and testing set contains all of the data of 500 samples. This means that 250 training samples have 5 repeats of 50 different dyeings and 500 testing samples means 10 repeats of 50 different dyeings. The detailed explanation of these applications and their results with comparison are given.

There are 50 different colors in the tables below. In the Table 5.1, Table 5.2 and Table 5.3; reflectance, CIELab and the output percentage values that form these colors are seen for 50 different dyeing respectively. In this thesis, there are 10 repeats of these 50 different dyeing. The total data set has 500 inputs for the network. Some of the data in these 500 inputs were used for training, the rest of them were used for testing.

Table 5.1 The reflectance values of 50 different dyeing between 400-500 nm.

	400	410	420	430	440	450	460	470	480	490	500
1	16,00	15,64	15,26	15,07	15,07	15,20	15,37	15,39	15,09	14,71	14,36
2	17,53	17,26	16,89	16,75	16,71	16,79	16,89	16,83	16,40	15,93	15,50
3	13,08	12,89	12,62	12,53	12,50	12,54	12,58	12,43	12,04	11,56	11,14
4	11,88	11,75	11,53	11,47	11,46	11,50	11,52	11,36	10,95	10,48	10,06
5	14,84	14,67	14,39	14,35	14,35	14,34	14,24	13,66	12,65	11,78	10,91
6	13,53	13,51	13,40	13,38	13,39	13,37	12,95	12,01	10,68	9,63	8,64
7	12,52	12,55	12,53	12,52	12,53	12,36	11,86	10,72	9,27	8,15	7,15
8	12,14	11,66	11,16	10,97	10,96	11,14	11,52	11,86	12,17	12,44	12,68
9	8,90	8,44	8,01	7,86	7,85	8,03	8,44	8,88	9,41	9,92	10,60
10	7,47	7,06	6,63	6,52	6,50	6,68	7,08	7,55	8,17	8,78	9,66
11	10,91	10,62	10,28	10,23	10,25	10,39	10,56	10,56	10,35	10,07	9,79
12	10,21	9,91	9,63	9,56	9,57	9,69	9,82	9,81	9,58	9,27	8,97
13	9,28	9,05	8,83	8,75	8,77	8,88	8,97	8,94	8,66	8,32	8,00
14	8,72	8,54	8,30	8,23	8,23	8,30	8,36	8,31	8,02	7,66	7,36
15	10,91	10,54	10,18	10,05	10,06	10,24	10,57	10,84	11,03	11,15	11,23
16	9,36	9,16	9,00	8,90	8,93	8,99	8,98	8,74	8,21	7,70	7,21
17	8,84	8,71	8,55	8,50	8,51	8,52	8,36	7,98	7,28	6,67	6,09
18	13,53	13,42	13,25	13,23	13,22	13,21	13,09	12,52	11,55	10,72	9,93
19	7,74	7,44	7,13	7,06	7,07	7,21	7,44	7,61	7,70	7,71	7,69
20	6,54	6,27	6,03	5,93	5,96	6,11	6,37	6,62	6,85	7,04	7,18
21	8,01	7,70	7,36	7,22	7,21	7,30	7,42	7,43	7,30	7,12	6,91
22	7,69	7,44	7,16	7,07	7,07	7,16	7,26	7,24	7,10	6,87	6,65
23	7,19	6,90	6,68	6,58	6,57	6,65	6,74	6,73	6,57	6,34	6,12
24	6,83	6,61	6,39	6,29	6,27	6,33	6,39	6,35	6,18	5,94	5,72
25	7,97	7,60	7,20	7,08	7,06	7,19	7,46	7,75	8,04	8,33	8,62
26	7,48	7,17	6,86	6,76	6,73	6,80	6,88	6,88	6,74	6,56	6,36
27	6,92	6,72	6,49	6,41	6,40	6,44	6,43	6,29	5,97	5,62	5,30
28	11,04	11,00	10,89	10,84	10,80	10,75	10,43	9,75	8,71	7,87	7,10
29	8,73	8,56	8,38	8,30	8,28	8,31	8,29	8,06	7,57	7,10	6,65
30	6,39	6,16	5,93	5,82	5,81	5,90	6,03	6,10	6,06	5,97	5,87
31	6,42	6,16	5,99	5,89	5,87	5,93	6,03	6,04	5,93	5,76	5,59
32	6,08	5,84	5,66	5,58	5,54	5,60	5,69	5,69	5,57	5,40	5,24
33	5,74	5,55	5,37	5,30	5,28	5,33	5,40	5,39	5,28	5,11	4,95
34	5,71	5,54	5,42	5,34	5,32	5,38	5,45	5,44	5,30	5,11	4,93
35	6,26	5,96	5,70	5,61	5,57	5,68	5,94	6,23	6,55	6,86	7,23
36	6,05	5,78	5,59	5,51	5,47	5,56	5,75	5,91	6,03	6,11	6,18
37	5,86	5,66	5,52	5,44	5,42	5,50	5,62	5,68	5,63	5,53	5,43
38	9,12	9,16	9,18	9,15	9,10	9,00	8,67	7,94	6,96	6,17	5,49
39	7,52	7,41	7,34	7,28	7,25	7,22	7,10	6,78	6,20	5,69	5,22
40	6,56	6,40	6,31	6,23	6,21	6,23	6,23	6,12	5,80	5,45	5,14
41	5,27	5,07	4,94	4,89	4,88	4,94	5,01	5,00	4,88	4,71	4,55
42	5,04	4,85	4,73	4,68	4,67	4,72	4,79	4,79	4,67	4,50	4,34
43	5,00	4,83	4,71	4,66	4,65	4,71	4,77	4,75	4,62	4,43	4,26
44	4,63	4,55	4,45	4,40	4,38	4,42	4,47	4,46	4,33	4,15	3,99
45	4,77	4,66	4,59	4,54	4,51	4,55	4,58	4,55	4,41	4,22	4,05
46	5,20	4,98	4,79	4,70	4,67	4,77	5,01	5,28	5,59	5,91	6,29
47	5,15	4,93	4,75	4,68	4,67	4,75	4,92	5,09	5,24	5,36	5,47
48	4,81	4,64	4,53	4,48	4,48	4,56	4,70	4,81	4,85	4,84	4,81
49	4,56	4,41	4,33	4,30	4,27	4,32	4,40	4,43	4,37	4,27	4,17
50	4,07	3,96	3,90	3,88	3,88	3,92	3,94	3,89	3,75	3,57	3,41

The continuum of Table 5.1. The reflectance values of 50 different dyeing between 510-600 nm.

	510	520	530	540	550	560	570	580	590	600
1	13,73	13,15	12,91	12,68	12,08	12,15	12,99	14,95	16,61	17,34
2	14,81	14,08	13,57	13,02	12,16	11,74	12,09	12,84	13,30	13,42
3	10,54	9,91	9,41	8,88	8,15	7,75	7,87	8,08	8,23	8,24
4	9,51	8,87	8,33	7,77	7,06	6,63	6,59	6,64	6,66	6,60
5	9,89	9,14	8,91	8,71	8,47	8,49	9,78	12,45	15,28	17,01
6	7,57	6,88	6,70	6,59	6,41	6,51	7,81	10,70	14,19	16,58
7	6,16	5,60	5,39	5,30	5,21	5,32	6,56	9,55	13,69	17,06
8	12,79	12,85	12,91	12,88	12,46	12,57	13,56	15,66	17,44	18,26
9	11,14	11,53	11,93	12,00	11,84	11,81	12,87	14,84	16,49	17,24
10	10,48	11,17	11,85	12,06	12,03	12,08	13,16	15,16	16,86	17,66
11	9,24	8,86	8,74	8,65	8,48	8,57	9,93	12,67	15,52	17,24
12	8,49	8,08	7,87	7,66	7,36	7,36	8,02	9,49	10,77	11,34
13	7,53	7,13	6,84	6,60	6,22	6,16	6,50	7,35	7,97	8,26
14	6,91	6,52	6,20	5,93	5,52	5,40	5,59	6,07	6,37	6,54
15	11,15	11,02	10,80	10,49	9,90	9,68	10,03	10,80	11,29	11,44
16	6,57	6,13	5,94	5,81	5,62	5,62	6,45	8,18	9,97	11,03
17	5,43	5,03	4,86	4,76	4,64	4,67	5,49	7,28	9,36	10,70
18	8,97	8,25	7,94	7,70	7,28	7,31	7,96	9,45	10,77	11,34
19	7,54	7,41	7,30	7,22	6,93	6,90	7,64	9,08	10,37	10,94
20	7,23	7,26	7,27	7,22	7,04	7,00	7,76	9,22	10,53	11,13
21	6,57	6,39	6,31	6,29	6,24	6,34	7,49	10,08	13,40	15,91
22	6,28	6,04	5,91	5,81	5,66	5,67	6,47	8,13	9,87	10,90
23	5,78	5,53	5,36	5,23	5,03	5,04	5,46	6,49	7,50	8,01
24	5,39	5,16	4,96	4,82	4,56	4,52	4,80	5,53	6,14	6,45
25	8,85	8,93	8,85	8,64	8,06	7,73	7,85	8,09	8,25	8,28
26	6,04	5,84	5,73	5,65	5,50	5,50	6,29	7,94	9,67	10,65
27	4,87	4,61	4,45	4,36	4,22	4,27	4,75	5,93	7,19	7,91
28	6,28	5,69	5,42	5,20	4,96	4,94	5,36	6,40	7,42	7,91
29	6,08	5,67	5,44	5,28	5,03	5,05	5,49	6,54	7,56	8,04
30	5,65	5,50	5,38	5,28	5,10	5,12	5,58	6,66	7,70	8,21
31	5,30	5,15	5,11	5,11	5,10	5,24	6,41	9,17	12,95	15,97
32	4,95	4,79	4,70	4,66	4,59	4,64	5,42	7,12	9,17	10,59
33	4,69	4,52	4,40	4,33	4,22	4,28	4,74	5,88	7,11	7,87
34	4,65	4,46	4,30	4,20	4,06	4,07	4,43	5,27	6,12	6,53
35	7,53	7,66	7,60	7,39	6,89	6,53	6,51	6,55	6,59	6,55
36	6,15	6,08	5,95	5,79	5,48	5,37	5,55	6,02	6,33	6,51
37	5,22	5,06	4,89	4,77	4,57	4,53	4,82	5,54	6,13	6,46
38	4,80	4,35	4,10	3,94	3,76	3,75	4,07	4,86	5,66	6,04
39	4,70	4,32	4,12	3,99	3,82	3,82	4,15	4,97	5,78	6,18
40	4,74	4,45	4,27	4,15	3,99	3,99	4,35	5,19	6,02	6,42
41	4,28	4,14	4,10	4,11	4,12	4,23	5,22	7,68	11,43	14,83
42	4,10	3,96	3,89	3,88	3,84	3,89	4,60	6,25	8,41	10,04
43	4,00	3,85	3,75	3,71	3,65	3,71	4,21	5,42	6,84	7,79
44	3,76	3,61	3,50	3,44	3,36	3,39	3,74	4,59	5,53	6,07
45	3,81	3,64	3,52	3,44	3,34	3,34	3,62	4,32	5,04	5,40
46	6,64	6,78	6,74	6,56	6,08	5,75	5,63	5,60	5,57	5,51
47	5,51	5,47	5,36	5,20	4,89	4,75	4,85	5,11	5,30	5,40
48	4,70	4,58	4,44	4,33	4,13	4,07	4,28	4,74	5,09	5,31
49	3,99	3,85	3,73	3,64	3,50	3,48	3,78	4,31	4,81	5,09
50	3,18	3,04	2,93	2,88	2,79	2,80	3,11	3,60	4,16	4,49

The continuum of Table 5.1. The reflectance values of 50 different dyeing between 610-700 nm.

	610	620	630	640	650	660	670	680	690	700
1	17,55	17,75	18,06	19,19	21,56	25,49	31,19	39,03	46,01	55,34
2	13,40	13,45	13,67	14,65	16,73	20,31	25,64	33,13	39,94	49,30
3	8,18	8,22	8,40	9,09	10,70	13,67	18,41	25,45	32,04	41,18
4	6,52	6,54	6,70	7,27	8,63	11,20	15,42	21,99	28,29	37,18
5	17,76	18,11	18,53	19,70	22,14	26,14	31,92	39,78	46,71	55,89
6	17,75	18,21	18,70	19,91	22,41	26,51	32,41	40,46	47,46	56,64
7	18,93	19,69	20,30	21,59	24,13	28,27	34,19	42,20	49,13	58,18
8	18,49	18,70	19,02	20,19	22,63	26,71	32,61	40,64	47,61	56,83
9	17,46	17,67	17,99	19,11	21,49	25,45	31,19	39,09	46,06	55,36
10	17,90	18,12	18,45	19,60	22,00	25,96	31,73	39,60	46,47	55,59
11	17,96	18,32	18,74	19,92	22,37	26,37	32,16	40,03	46,87	55,92
12	11,55	11,72	11,98	12,90	14,87	18,32	23,55	30,97	37,67	46,79
13	8,36	8,46	8,75	9,40	11,03	13,96	18,59	25,46	31,84	40,67
14	6,57	6,63	6,80	7,39	8,75	11,31	15,51	22,00	28,22	36,98
15	11,45	11,51	11,77	12,63	14,59	17,99	23,17	30,54	37,23	46,36
16	11,46	11,70	12,02	12,95	14,93	18,38	23,61	30,98	37,62	46,62
17	11,29	11,60	11,96	12,87	14,85	18,30	23,56	31,02	37,72	46,84
18	11,56	11,74	12,00	12,92	14,92	18,40	23,69	31,18	37,93	47,13
19	11,16	11,33	11,62	12,49	14,43	17,82	23,01	30,42	37,12	46,23
20	11,36	11,54	11,81	12,73	14,67	18,06	23,21	30,55	37,16	46,21
21	17,27	17,83	18,36	19,54	21,93	25,85	31,54	39,34	46,15	55,17
22	11,33	11,60	11,95	12,85	14,84	18,26	23,37	30,61	37,16	46,16
23	8,23	8,40	8,75	9,39	11,00	13,89	18,43	25,14	31,42	40,18
24	6,58	6,69	6,87	7,47	8,87	11,47	15,72	22,21	28,44	37,21
25	8,24	8,28	8,47	9,16	10,78	13,78	18,55	25,63	32,23	41,37
26	11,01	11,28	11,56	12,48	14,46	17,93	23,21	30,70	37,48	46,70
27	8,16	8,37	8,71	9,35	10,99	13,94	18,61	25,51	31,95	40,92
28	8,13	8,28	8,62	9,25	10,88	13,82	18,49	25,44	31,91	40,90
29	8,25	8,41	8,75	9,38	11,03	13,95	18,58	25,43	31,81	40,64
30	8,43	8,60	8,94	9,59	11,22	14,14	18,78	25,59	31,92	40,67
31	17,64	18,34	18,91	20,15	22,59	26,69	32,52	40,44	47,24	56,26
32	11,27	11,61	11,96	12,91	14,89	18,35	23,58	31,00	37,61	46,71
33	8,16	8,40	8,75	9,40	10,99	13,92	18,51	25,34	31,68	40,53
34	6,75	6,91	7,21	7,77	9,18	11,79	16,02	22,49	28,64	37,28
35	6,50	6,55	6,71	7,29	8,63	11,19	15,39	21,86	28,05	36,82
36	6,56	6,64	6,81	7,39	8,73	11,30	15,47	21,91	28,08	36,81
37	6,60	6,73	6,92	7,52	8,89	11,46	15,62	22,02	28,12	36,74
38	6,25	6,39	6,58	7,15	8,47	10,99	15,14	21,61	27,84	36,66
39	6,39	6,54	6,73	7,31	8,64	11,18	15,31	21,73	27,88	36,63
40	6,64	6,79	7,01	7,61	8,98	11,58	15,78	22,23	28,40	37,09
41	16,94	17,85	18,54	19,81	22,23	26,24	31,94	39,72	46,42	55,33
42	10,91	11,30	11,68	12,62	14,54	17,90	22,95	30,15	36,62	45,55
43	8,22	8,48	8,84	9,53	11,17	14,10	18,69	25,50	31,78	40,54
44	6,34	6,53	6,78	7,34	8,66	11,13	15,15	21,39	27,38	35,90
45	5,59	5,72	5,90	6,43	7,65	9,98	13,88	20,04	26,06	34,64
46	5,46	5,49	5,62	6,11	7,26	9,51	13,29	19,33	25,31	33,89
47	5,42	5,48	5,62	6,10	7,24	9,46	13,19	19,16	25,07	33,59
48	5,40	5,49	5,65	6,14	7,27	9,41	13,01	18,78	24,50	32,79
49	5,21	5,31	5,46	5,95	7,07	9,27	12,99	18,98	24,91	33,42
50	4,66	4,77	4,92	5,36	6,33	8,30	11,68	17,23	22,88	31,14

Table 5.2 The CIELab values of 50 different dyeing.

	L	a	b	c	h	X	Y	Z
1	44,85	8,77	-1,73	8,94	348,86	15,11	14,44	16,27
2	43,76	2,75	-7,08	7,60	291,21	13,38	13,67	17,91
3	36,37	1,24	-9,45	9,53	277,47	8,87	9,20	13,31
4	33,85	-0,06	-10,86	10,86	269,67	7,52	7,94	12,17
5	40,88	17,65	-5,54	18,49	342,58	13,77	11,79	14,91
6	38,10	23,31	-7,05	24,35	343,18	12,80	10,15	13,55
7	36,62	28,33	-6,71	29,12	346,68	12,60	9,33	12,41
8	44,73	8,41	7,62	11,35	42,17	14,96	14,35	12,27
9	42,98	8,09	13,97	16,14	59,91	13,69	13,15	9,05
10	42,90	8,55	18,64	20,50	65,37	13,71	13,09	7,64
11	40,34	16,34	3,23	16,66	11,19	13,20	11,45	11,10
12	36,20	9,47	-1,69	9,62	349,89	9,78	9,11	10,34
13	32,86	6,27	-4,69	7,83	323,19	7,74	7,47	9,43
14	30,60	4,11	-6,59	7,76	301,92	6,53	6,48	8,81
15	39,37	2,33	1,28	2,65	28,79	10,61	10,87	11,21
16	33,48	14,78	-3,53	15,20	346,56	8,99	7,76	9,40
17	31,64	18,13	-4,66	18,72	345,57	8,47	6,93	8,77
18	36,82	11,51	-9,65	15,02	320,03	10,38	9,44	13,70
19	34,86	9,07	4,00	9,91	23,80	9,02	8,43	7,86
20	34,72	8,97	7,89	11,94	41,35	8,94	8,36	6,75
21	36,47	20,76	6,87	21,87	18,30	11,42	9,25	7,83
22	33,02	13,62	1,60	13,71	6,69	8,63	7,55	7,65
23	30,20	9,48	-1,25	9,56	352,47	6,89	6,32	7,10
24	28,27	7,11	-3,19	7,80	335,81	5,88	5,56	6,75
25	34,81	-0,47	3,49	3,52	97,60	7,92	8,40	7,98
26	32,53	13,59	2,09	13,75	8,74	8,39	7,32	7,28
27	28,65	12,74	-2,65	13,02	348,26	6,55	5,70	6,77
28	30,96	12,37	-12,51	17,59	314,69	7,52	6,64	10,96
29	30,69	10,48	-6,08	12,11	329,87	7,20	6,52	8,70
30	30,22	9,32	1,65	9,47	10,01	6,88	6,33	6,38
31	34,76	24,43	9,52	26,21	21,29	10,92	8,38	6,37
32	30,81	16,77	4,20	17,29	14,07	7,91	6,57	6,00
33	28,30	12,08	1,18	12,14	5,58	6,35	5,57	5,70
34	27,11	9,35	-1,06	9,41	353,52	5,64	5,13	5,75
35	31,75	-2,07	4,29	4,76	115,78	6,42	6,98	6,38
36	29,61	2,89	1,67	3,34	30,07	6,02	6,08	6,11
37	28,01	6,65	-0,31	6,66	357,32	5,74	5,46	5,93
38	27,08	12,53	-13,59	18,49	312,68	5,91	5,12	9,10
39	26,88	10,94	-8,33	13,76	322,71	5,69	5,05	7,47
40	27,16	9,92	-4,40	10,85	336,06	5,71	5,15	6,57
41	32,57	26,71	10,49	28,70	21,43	9,98	7,34	5,27
42	28,96	18,74	5,32	19,48	15,85	7,28	5,82	5,05
43	27,00	14,38	2,10	14,53	8,31	6,04	5,09	5,02
44	24,99	11,07	0,10	11,07	0,53	5,02	4,41	4,71
45	24,42	9,20	-1,48	9,32	350,86	4,68	4,23	4,84
46	29,55	-3,03	4,77	5,65	122,39	5,48	6,05	5,38
47	27,54	1,30	1,95	2,34	56,40	5,12	5,29	5,24
48	26,00	4,40	0,62	4,45	7,98	4,84	4,75	4,97
49	24,45	6,91	-0,50	6,93	355,88	4,52	4,24	4,65
50	22,11	8,77	-1,89	8,97	347,81	3,93	3,55	4,14

Table 5.3 The output percentage values of 50 different dyeing.

	Procion Crimson H-EXL (%)	Procion Navy H-EXL (%)	Procion Yellow H-E4R (%)
1	0,25	0,25	0,25
2	0,25	0,50	0,25
3	0,25	0,75	0,25
4	0,25	1,00	0,25
5	0,50	0,25	0,25
6	0,75	0,25	0,25
7	1,00	0,25	0,25
8	0,25	0,25	0,50
9	0,25	0,25	0,75
10	0,25	0,25	1,00
11	0,50	0,50	0,50
12	0,50	0,25	0,50
13	0,50	0,75	0,50
14	0,50	1,00	0,50
15	0,25	0,50	0,50
16	0,75	0,50	0,50
17	1,00	0,50	0,50
18	0,50	0,50	0,25
19	0,50	0,50	0,75
20	0,50	0,50	1,00
21	0,75	0,75	0,75
22	0,75	0,25	0,75
23	0,75	0,50	0,75
24	0,75	1,00	0,75
25	0,25	0,75	0,75
26	0,50	0,75	0,75
27	1,00	0,75	0,75
28	0,75	0,75	0,25
29	0,75	0,75	0,50
30	0,75	0,75	1,00
31	1,00	1,00	1,00
32	1,00	0,25	1,00
33	1,00	0,50	1,00
34	1,00	0,75	1,00
35	0,25	1,00	1,00
36	0,50	1,00	1,00
37	0,75	1,00	1,00
38	1,00	1,00	0,25
39	1,00	1,00	0,50
40	1,00	1,00	0,75
41	1,25	1,25	1,25
42	1,25	0,25	1,25
43	1,25	0,50	1,25
44	1,25	0,75	1,25
45	1,25	1,00	1,25
46	0,25	1,25	1,25
47	0,50	1,25	1,25
48	0,75	1,25	1,25
49	1,00	1,25	1,25
50	1,50	1,50	1,50

5.2 Applications of Artificial Neural Networks

As mentioned in the theoretical part; there are some methods used in artificial neural network in order to train a system. In this project, two of these methods were used; RBF and MLP. Both of these methods have some advantages and disadvantages. By using these two methods with the same data; we have an opportunity of comparing them. Also by changing various parameters of these methods, we observed how these parameters affect the results.

5.2.1 RBF Applications

In Matlab, there are four functions related to RBF. These functions are newrbe, newrb, newgrnn, newpnn. Applications were done with all of them. After observing the results, the two of them were found more successful comparing to the others. These two methods are newrbe and newrb. In the rest of the thesis, we only employed these two methods and their results are given in the result part.

While implementing newrbe, we did the following changes and explained as below.First of all, the most important parameter in the RBF applications is spread of the RBF. Because of this we focused on finding the exact spread value. For this reason a network with changing inputs related to data types and three outputs was formed. The results were observed while different spread values were given to the system. These spread values were given to the system manually. In this application it was recognized that in a spread value red color concentration was more successful than yellow and blue in the training. In another spread value yellow color concentration was more successful than red and blue in the training. By taking into consideration this result, it was understood that it is too difficult to train all of the color concentrations with the same spread value.

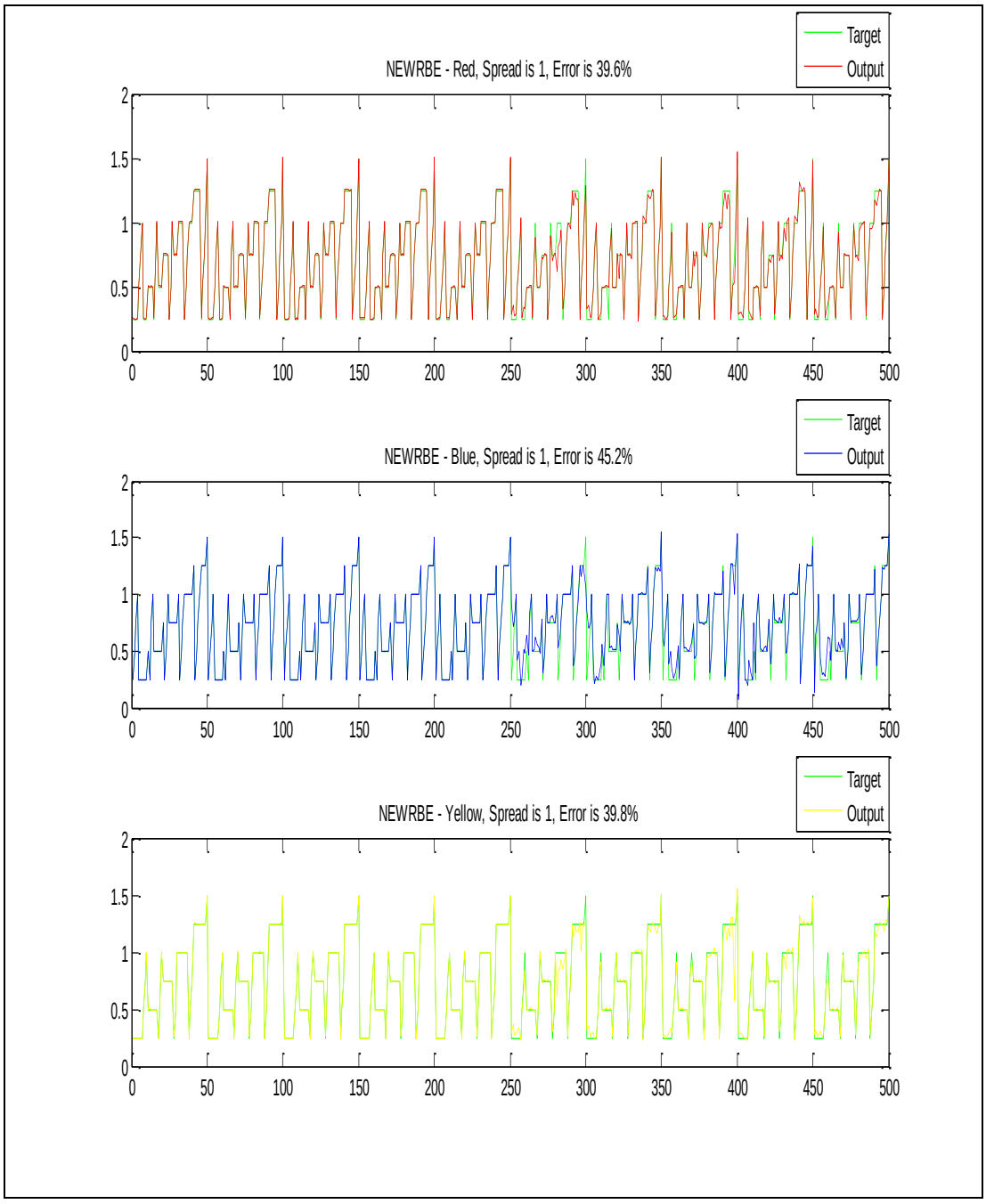

Figure 5.1 Training with a spread value of 1

As it can be seen from Figure 5.1 the system was trained with a spread value of 1. With this spread value, all of the outputs gave different error percentages in the result.

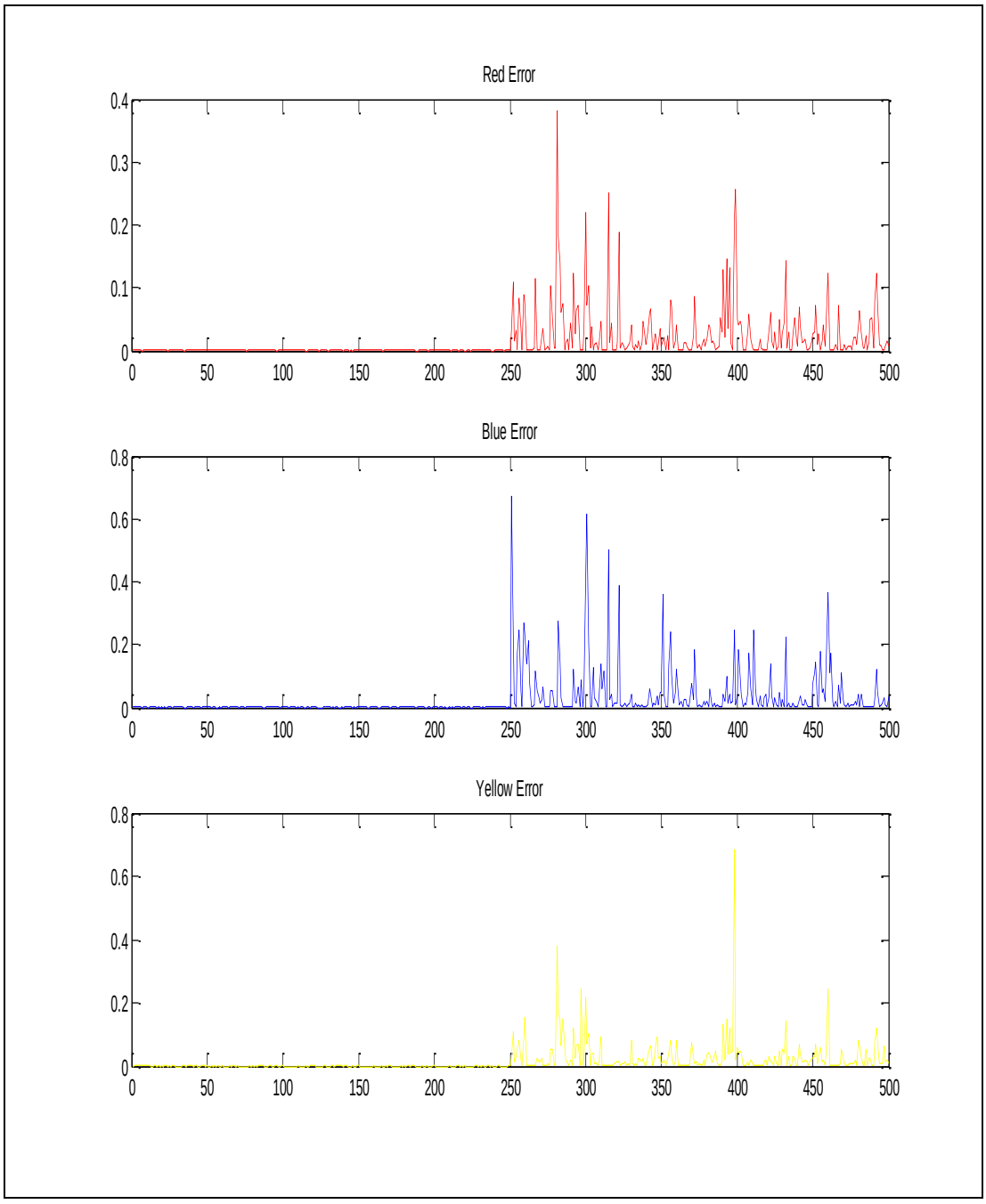

Figure 5.2 The error values of Figure 5.1

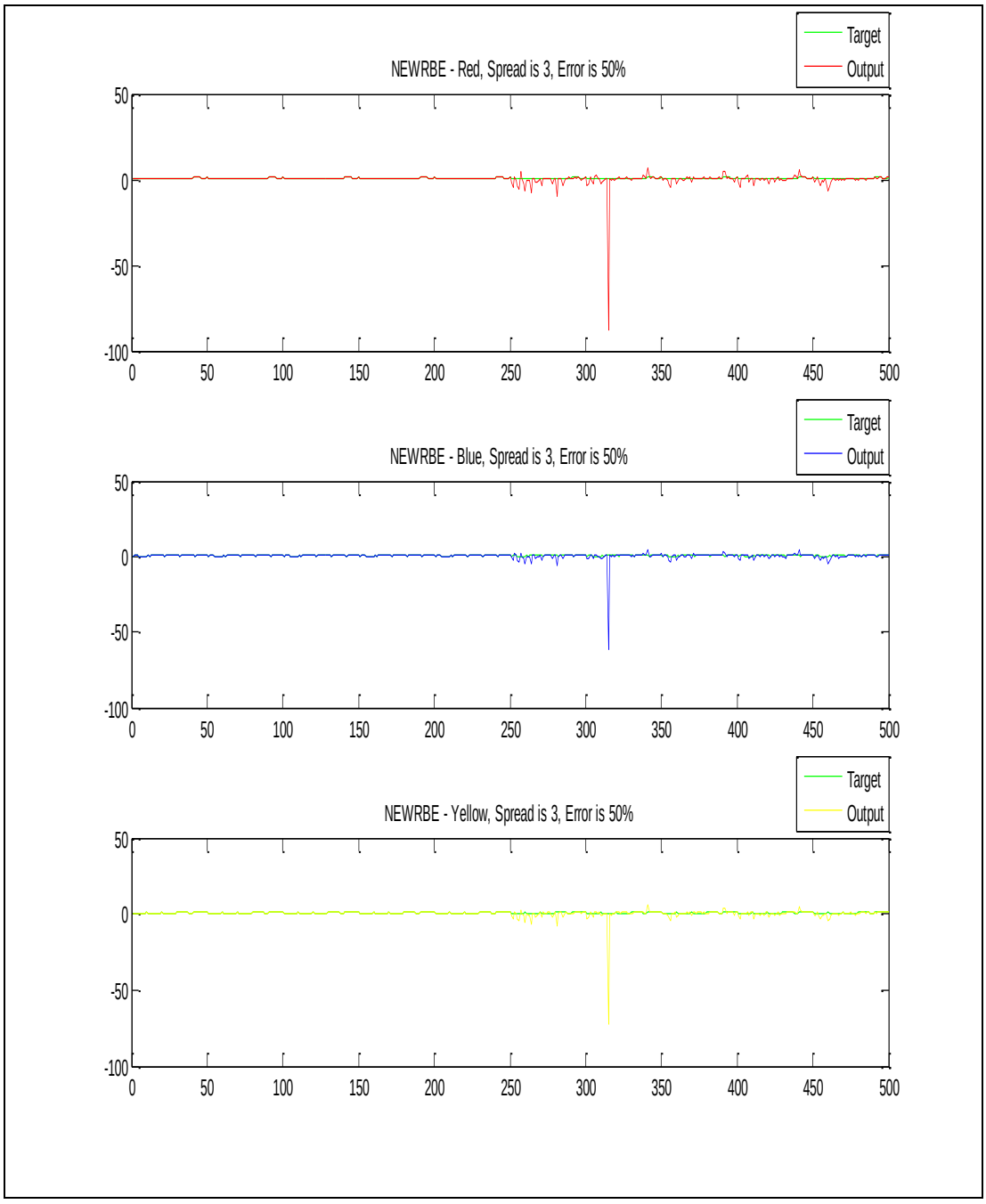

Figure 5.3 Training with a spread value of 3.

For various spread values the error value of the system also changes as seen in Figure 5.2 and Figure 5.4. This point out that spread is a very important parameter in the training of the RBF. There is no doubt, selecting the right spread value decreases the error.

It can be seen easily that every output needs a different spread value from Figure 5.1 and Figure 5.3. Because of this, this will be tried in the next application.

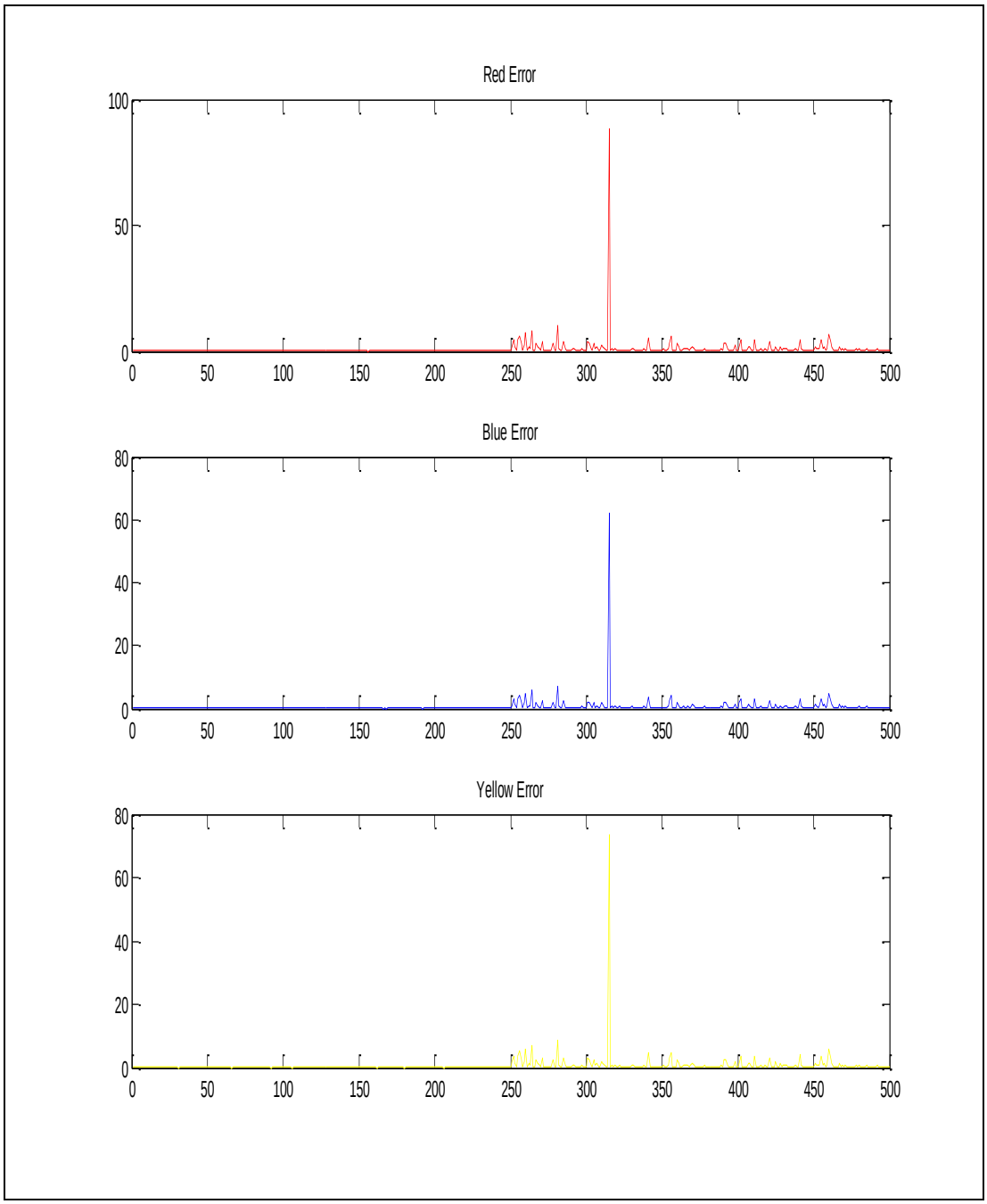

Figure 5.4 The error values of Figure 5.3

As a result, it was decided to train the colors red, blue and yellow with different spreads. Training these color values with different spreads is only possible by setting up different nets for them. In order to realize this, three different networks were set and for these networks three different spread values were given manually. By this way it was noticed that the networks were trained much better compared to a single network. Here, the disadvantage of the system is to find the correct spread value after many trials.

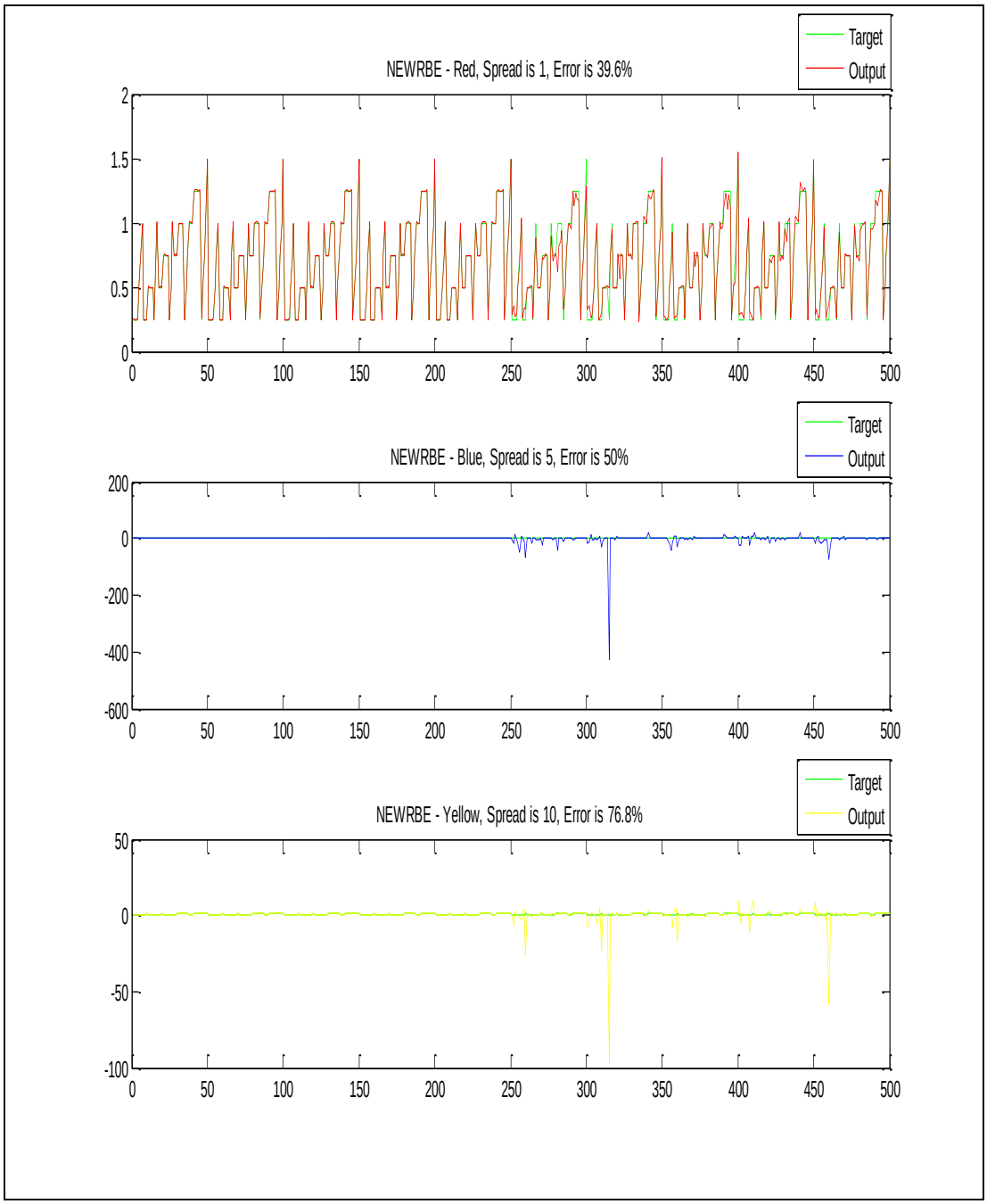

Figure 5.5 Training with different nets and different spread values

The system was trained with three different networks and three different spreads. As it can be seen from the Figure 5.5, red output was trained with spread 1, blue output was trained with spread 5 and yellow output was trained with spread 10. These spreads were selected manually. Selecting the right spread values is very important in the training process and it is very difficult to select the right spread values manually. Because of this, in the next application the spread values will be tried to select automatically.

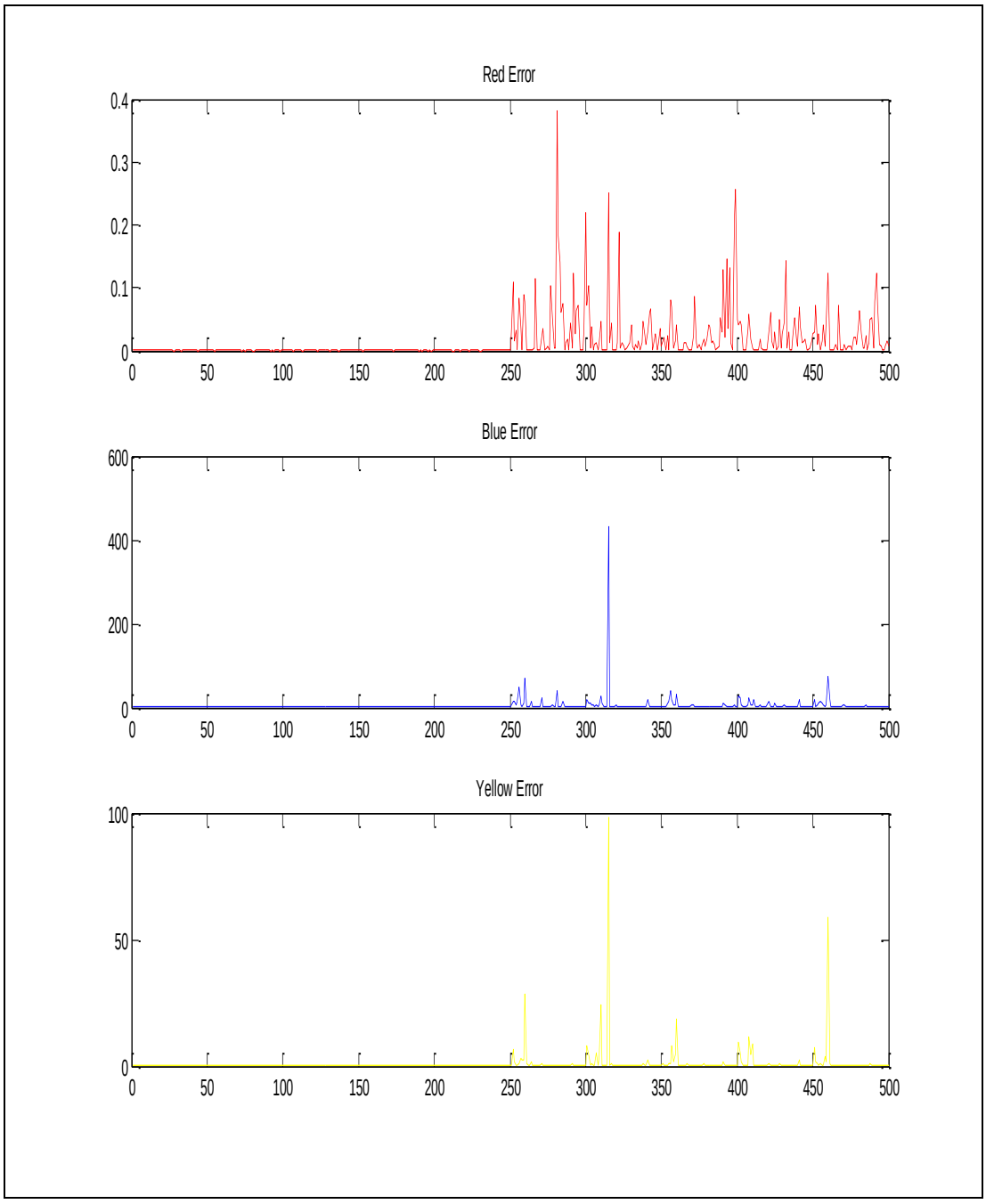

Figure 5.6 The error values of Figure 5.5

The error values of Figure 5.5 are seen in Figure 5.6.

For overcoming the problem of selecting the spread value manually, a subroutine was added to the program that calculates all of the spread values in a specified range. All of the values in a specified range were used as spread values and the spread value with minimum error for the nets was calculated and selected automatically. In this way, because of finding the exact spread values, all of the three nets had very successful and nearly the same performances.

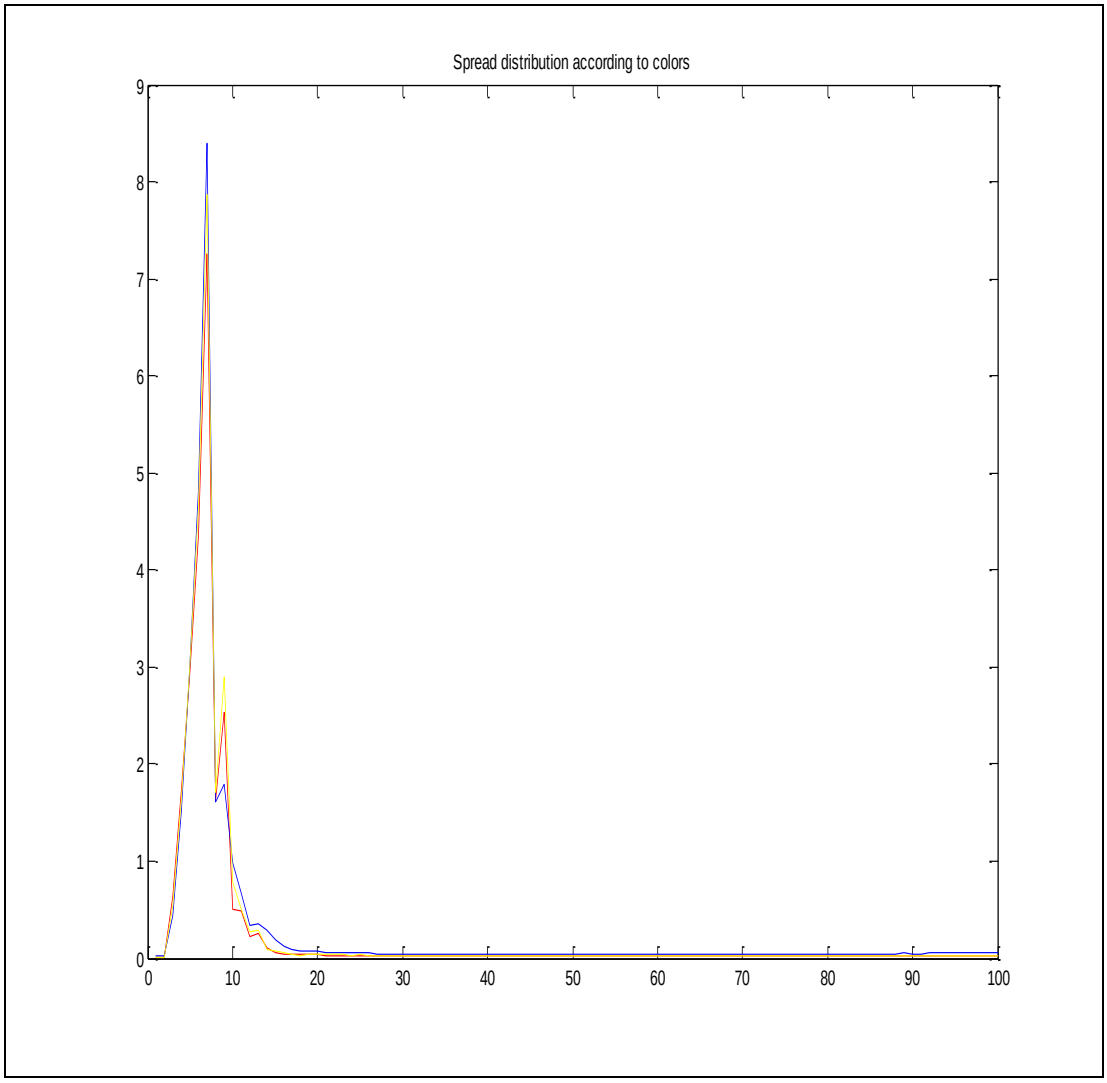

Figure 5.7 Spread distribution according to colors

According to Figure 5.7, the spread values between 0-100 are given with respect to the error values.

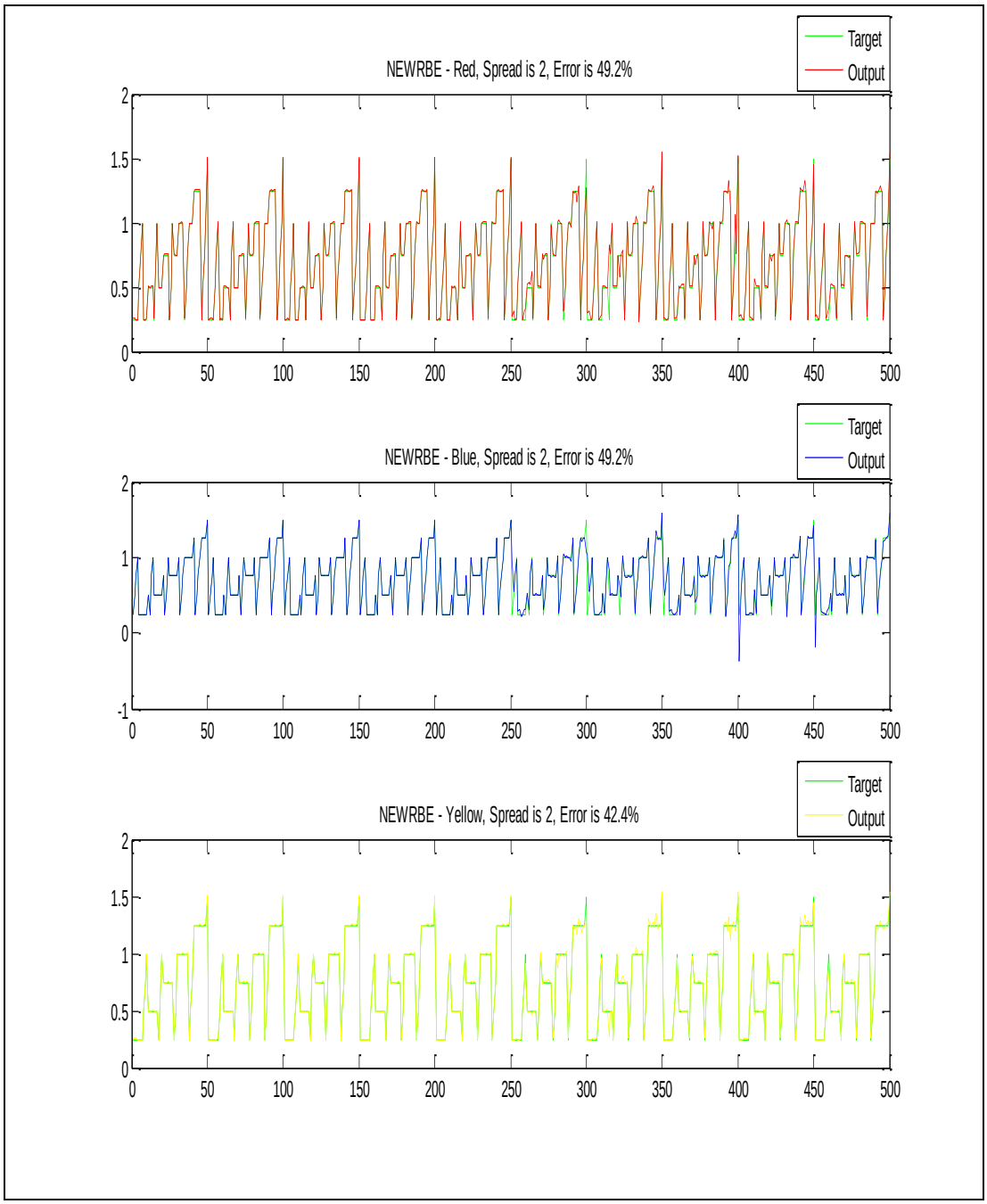

Figure 5.8 Automatic spread values for three different networks.

As shown in Figure 5.8 the spread values were given automatically from the graph of Figure 5.7. These spread values were determined by automatic spread calculation subroutine. This subroutine works by looking at the minimum error values and the spreads that correspond to these minimum error values are used in the training. In the calculation, the error values were found close to each other. With this obtained spread values, the error values have been successfully reduced.

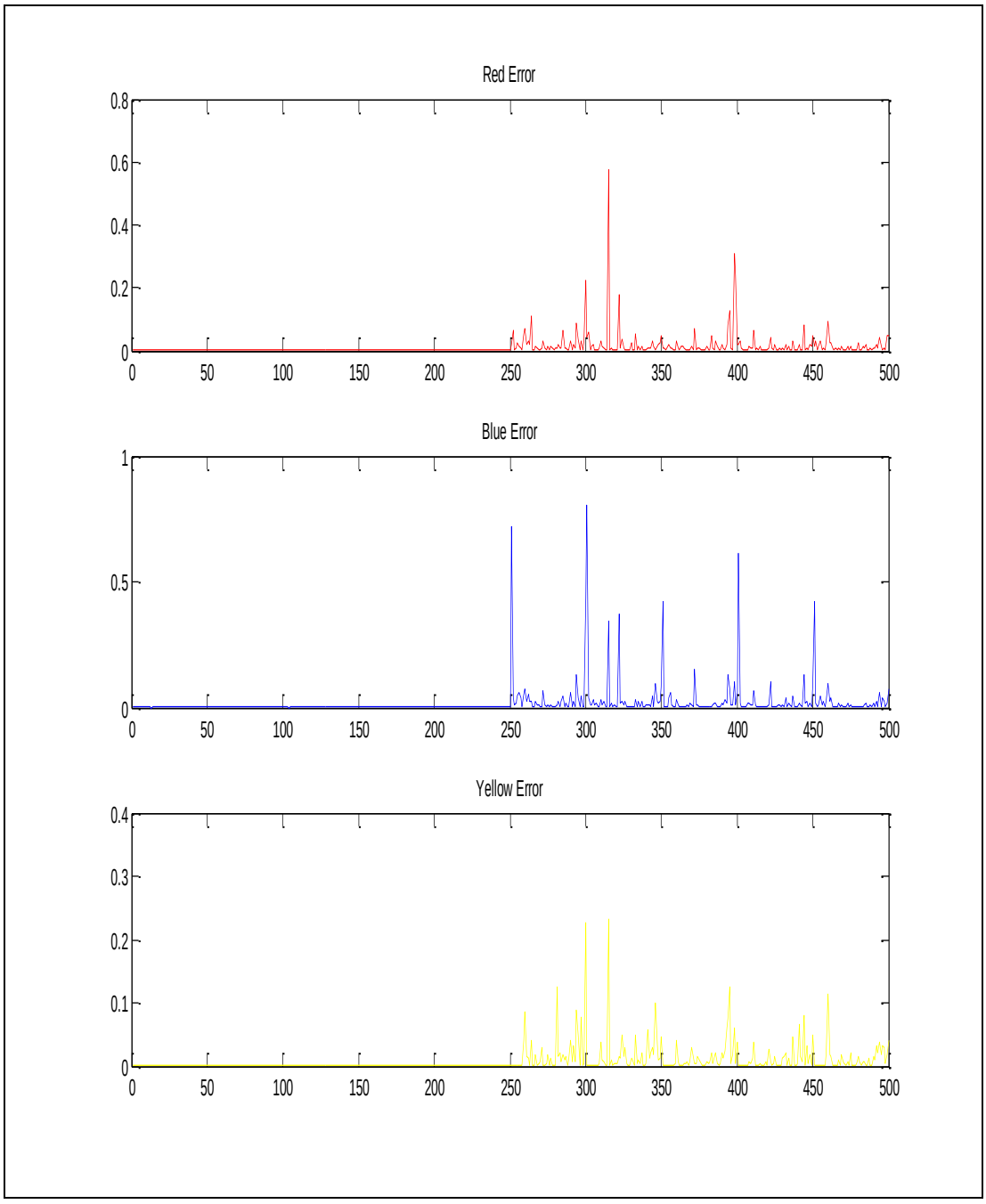

Figure 5.9 The error values for automatic spreads

As it can be seen from the Figure 5.9, the amplitude of error values were decreased by selecting automatic spread selection.

In addition to the methods above, we did some operations on the data. One of these operations was normalization. For calculating the normalization value, we generated a minimum vector which consists of the minimum values of each column. Also we generated a maximum vector which consists of the maximum values of each column. After these operations, we subtracted the minimum values from the original values and divide the result to the maximum values. By this way we converted all the values to the range of 0-1. In this operation performance was not affected so much.

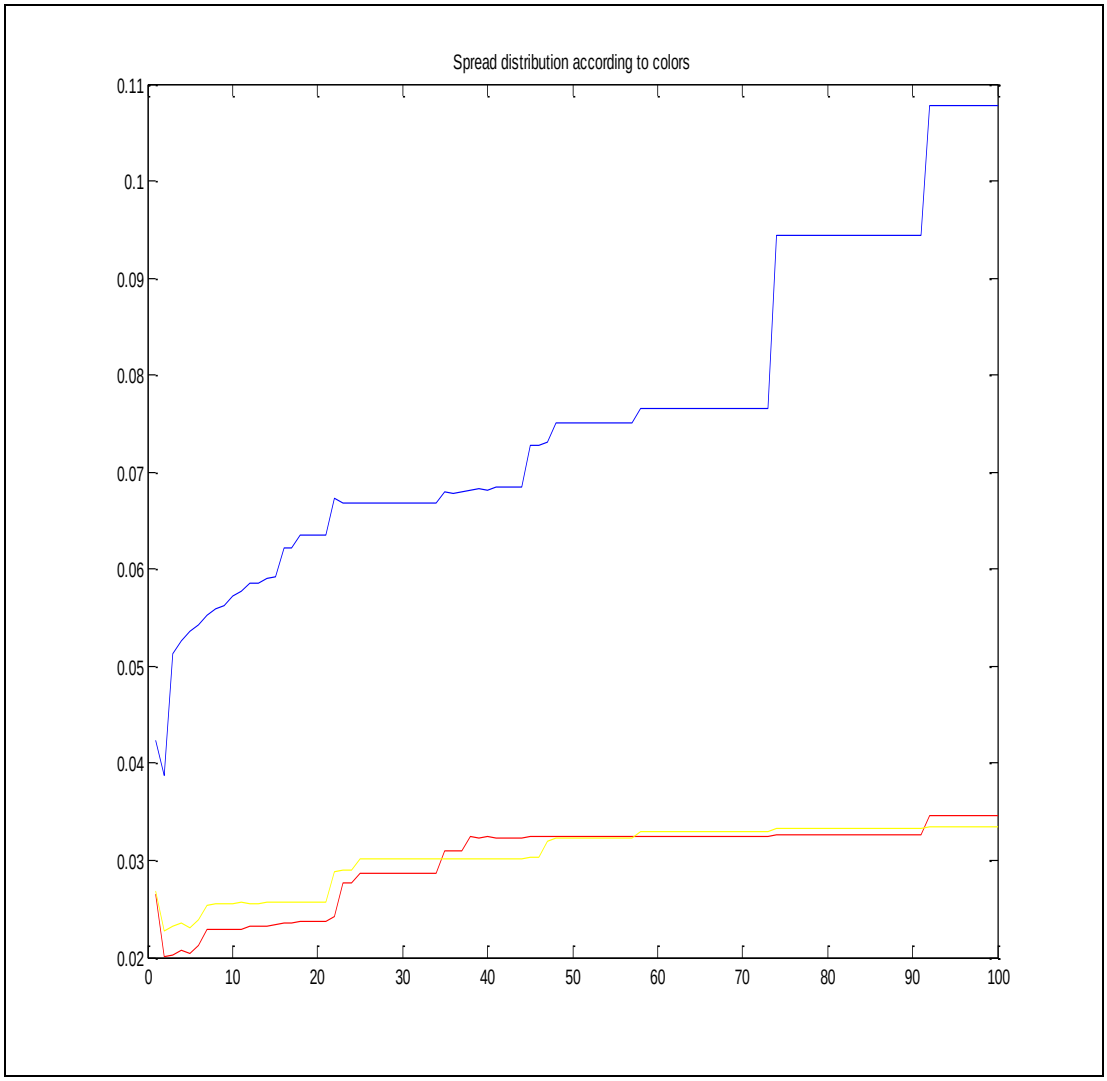

Figure 5.10 Spread distribution according to colors

According to Figure 5.10, the spread values between 0-100 are given with respect to the error values.

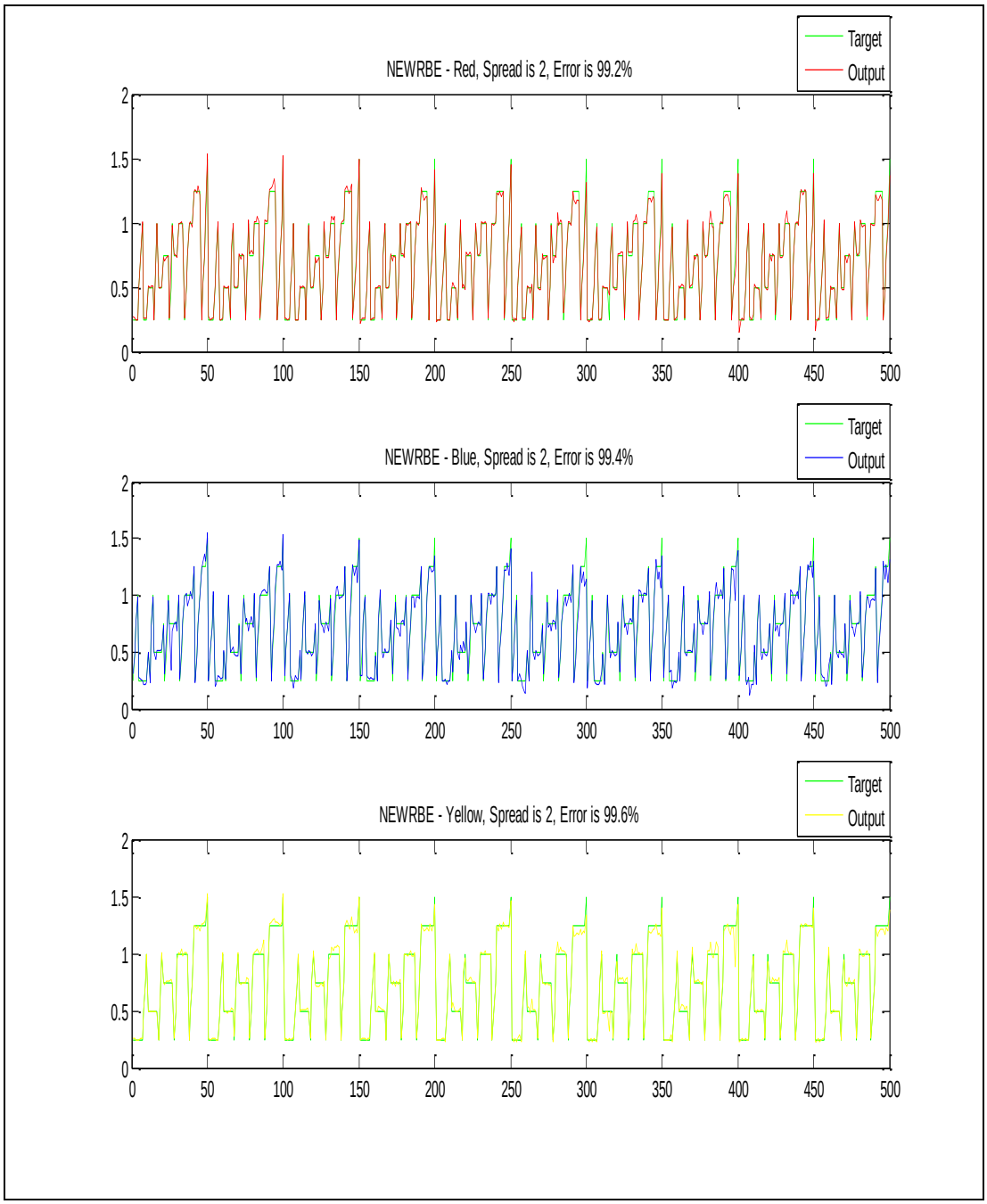

Figure 5.11 The output distribution after normalization

In Figure 5.11 the training with normalized data is shown. As it can be seen, after the normalization process, the error rate increased to a higher percentage.

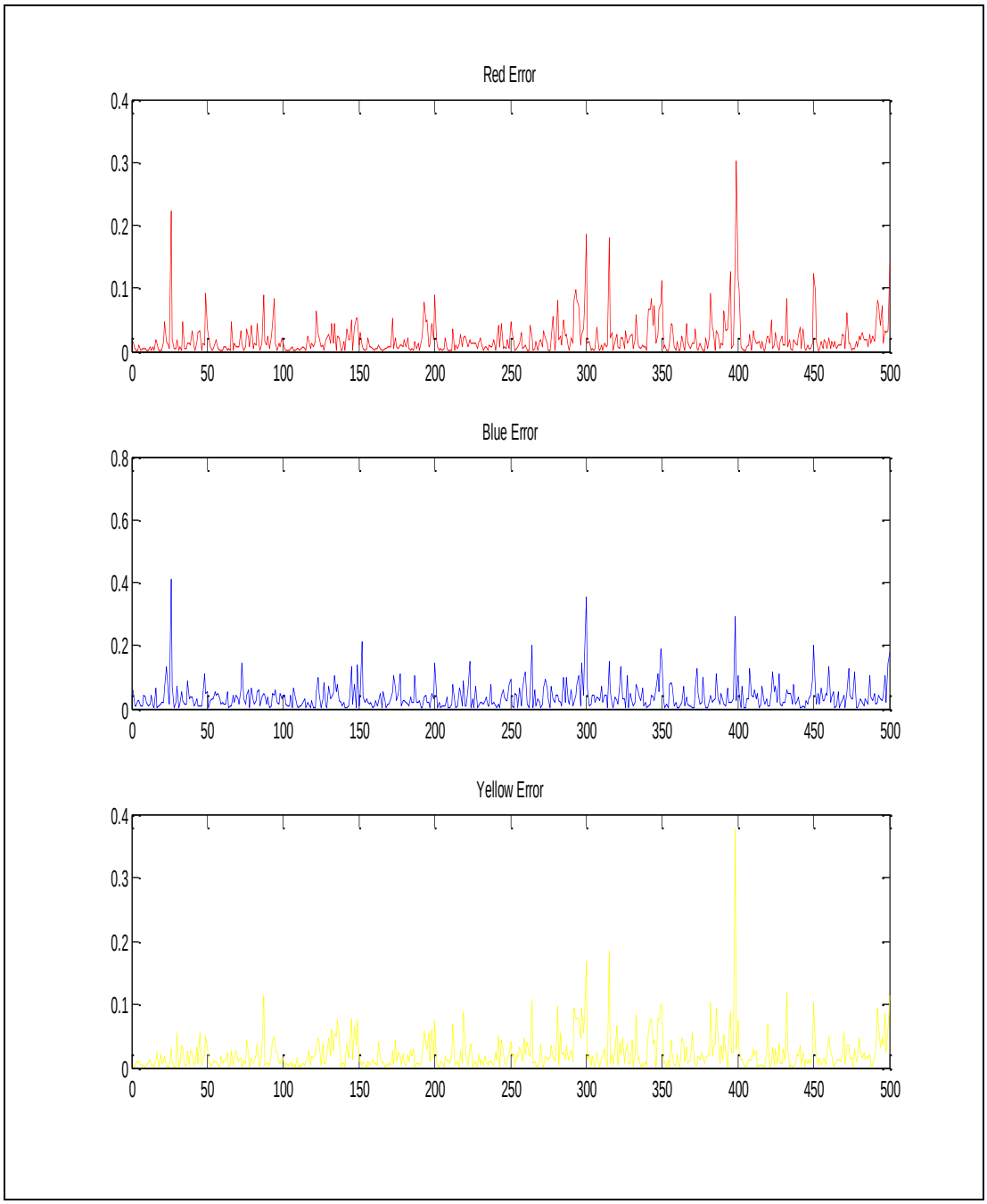

Figure 5.12 The error values for automatic spreads in normalization

The error values of Figure 5.11 are seen in Figure 5.12.

In addition to normalization of input data, the data were shuffled in a random manner. Our data was previously in a sequential order. Because of this, when we gave the data to the system directly, it was overtraining and therefore lost the ability of generalization. Then, when we gave a new data to the system, the system was not able to learn it in a good way. As a solution, we gave the data in a random order. By this way, the system not only learnt the data we gave, but also it recognized the new data given. Then, we can say that training is more successful randomly ordered data.

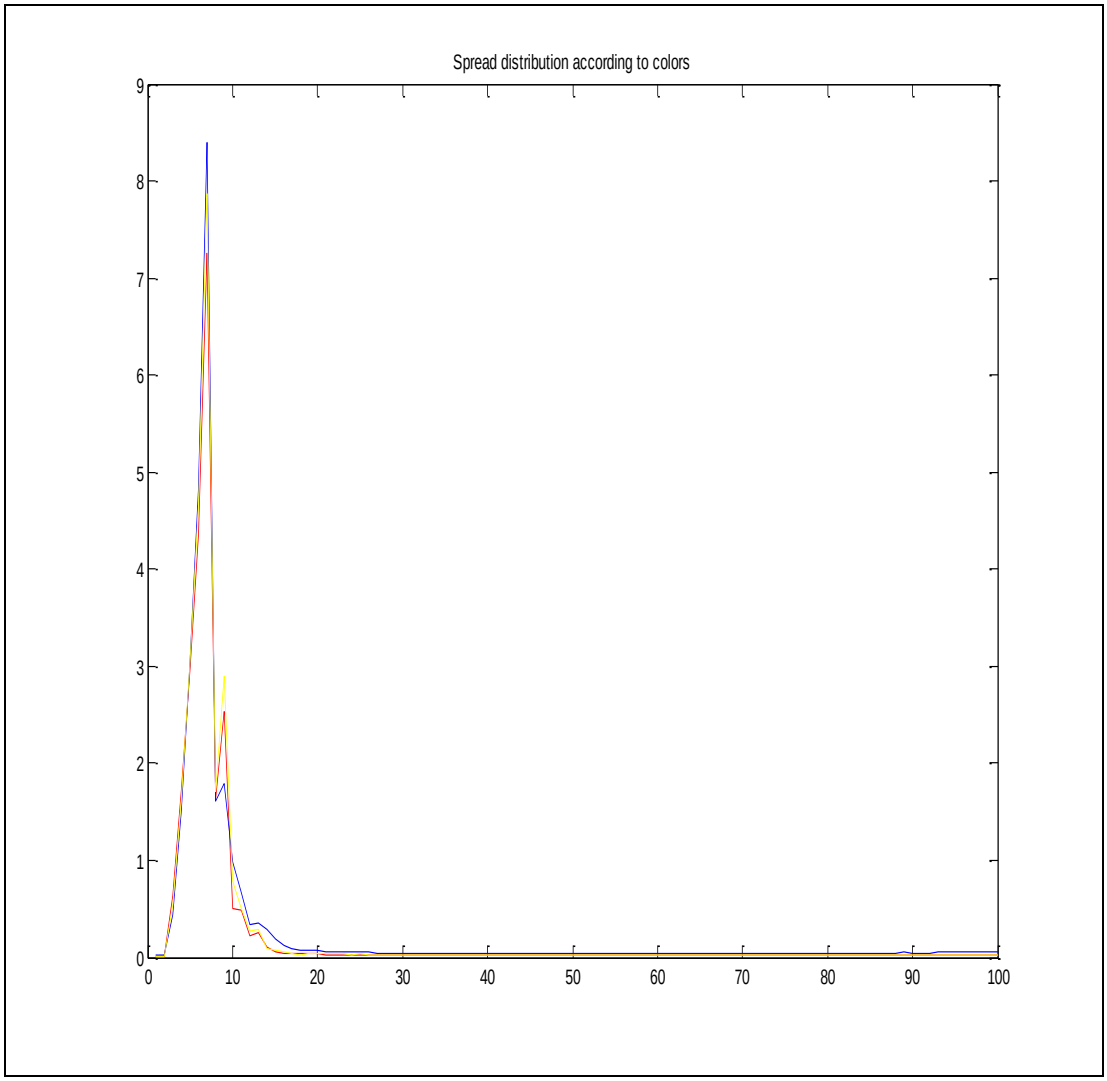

Figure 5.13 Spread distribution according to colors

The Figure 5.13 shows the spread distribution for randomly ordered data.

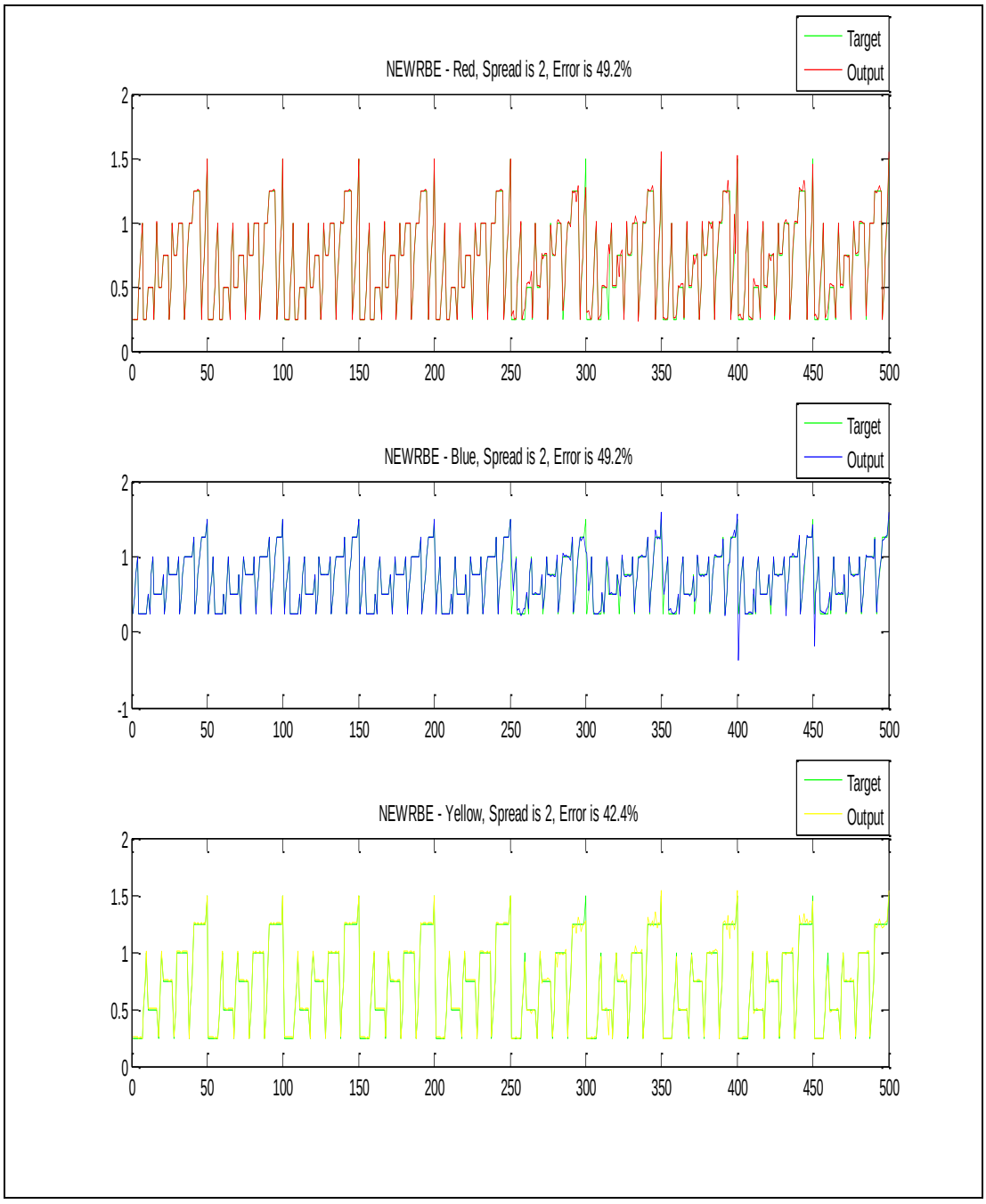

Figure 5.14 The output distribution after sorting randomly

Sorting the data and giving it to the system randomly has a positive effect on the system. However it did not change the result too much as seen in Figure 5.14.

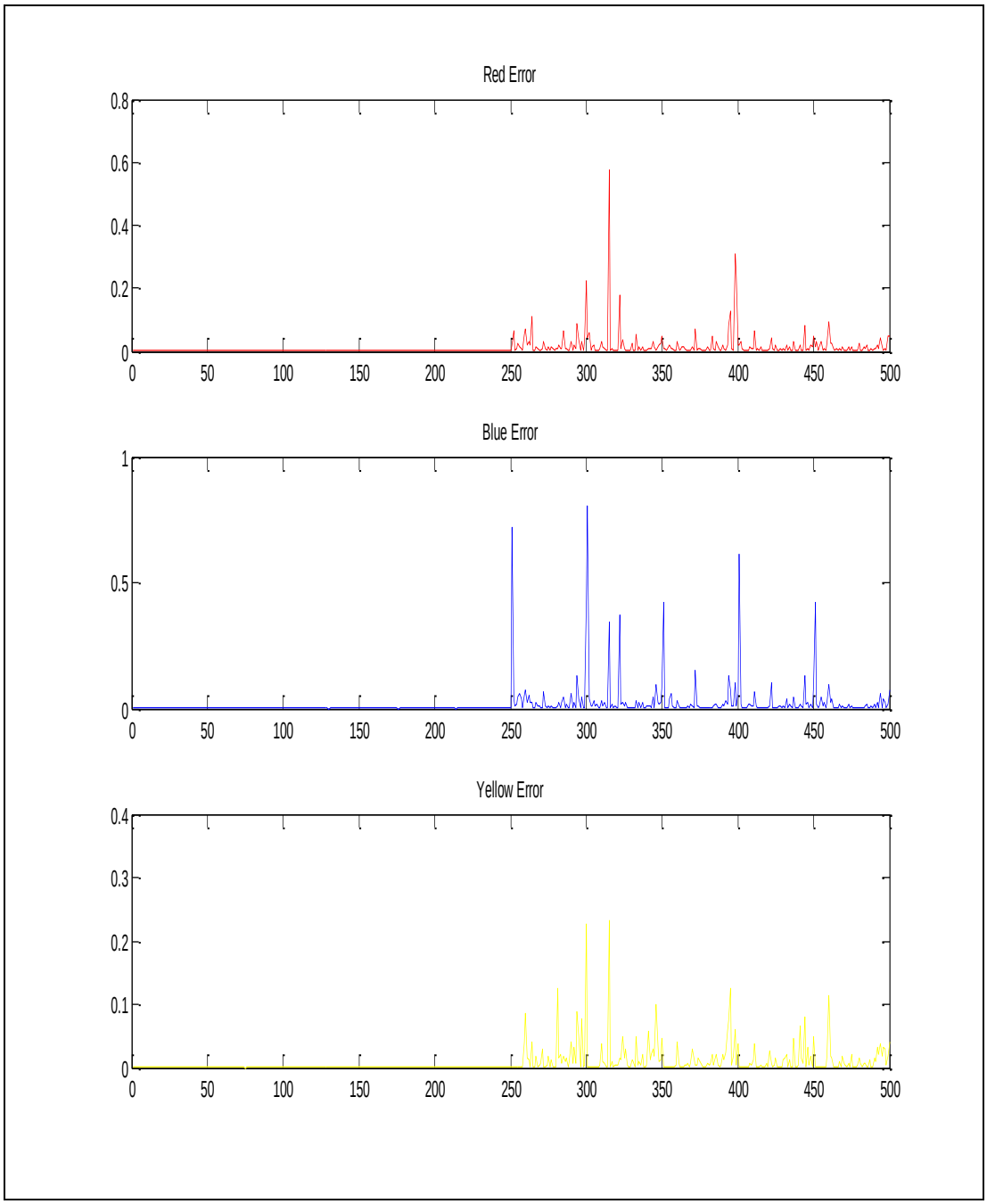

Figure 5.15 The error values for randomly sorting process

As it can be seen from the Figure 5.15, the amplitude of error values was decreased slightly.

The outputs that are defined as red, blue and yellow of the net have definite values. After training the net, it was seen that the network output did approximate the actual output values but not exactly the same. Because of this, the error rate was high. In order to overcome this problem, the obtained outputs were rounded to the closest output value. By this way, the error rate of the system was decreased in an acceptable value.

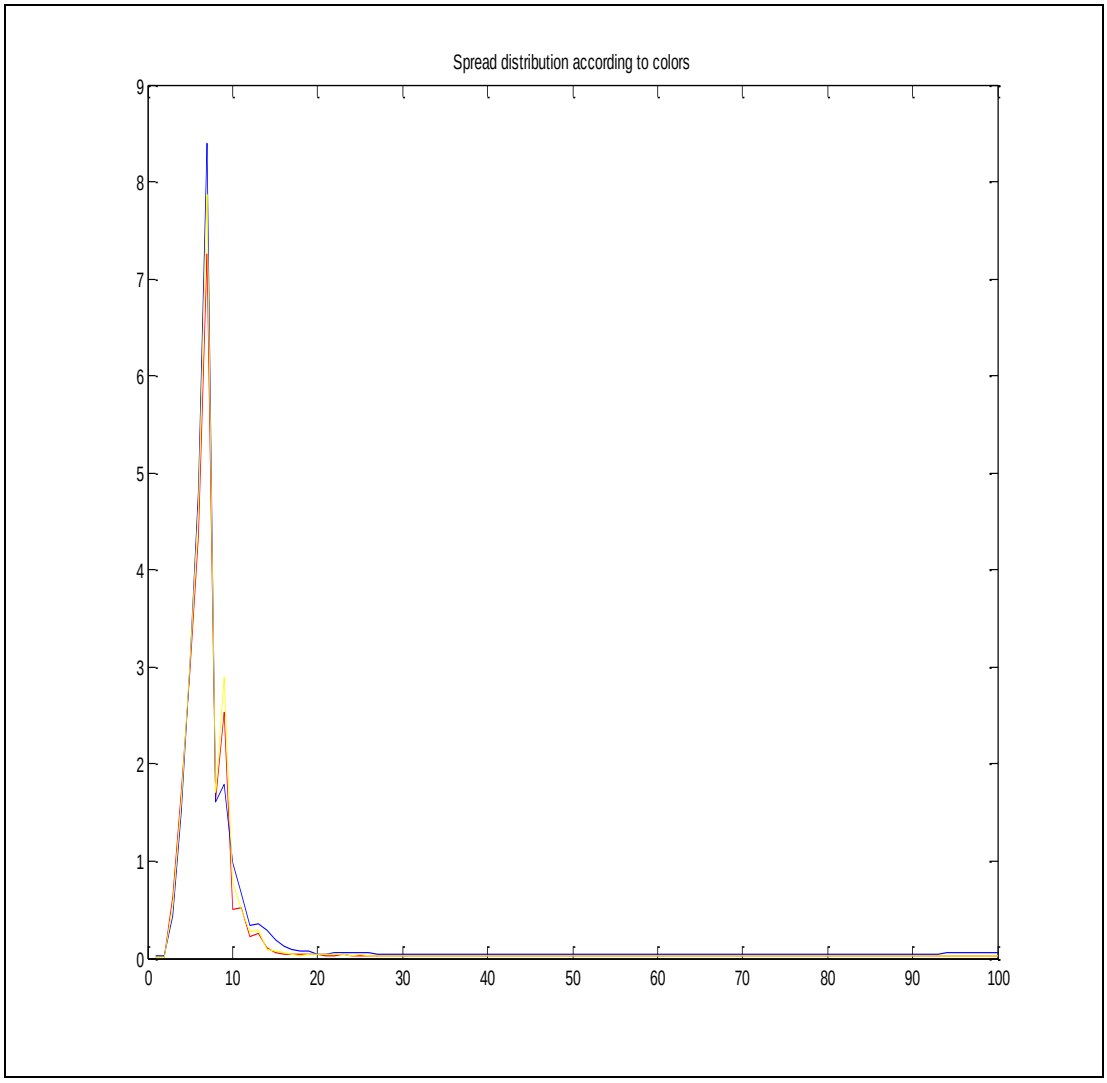

Figure 5.16 Spread distribution according to colors

According to Figure 5.16, the spread values between 0-100 are given with respect to the error values.

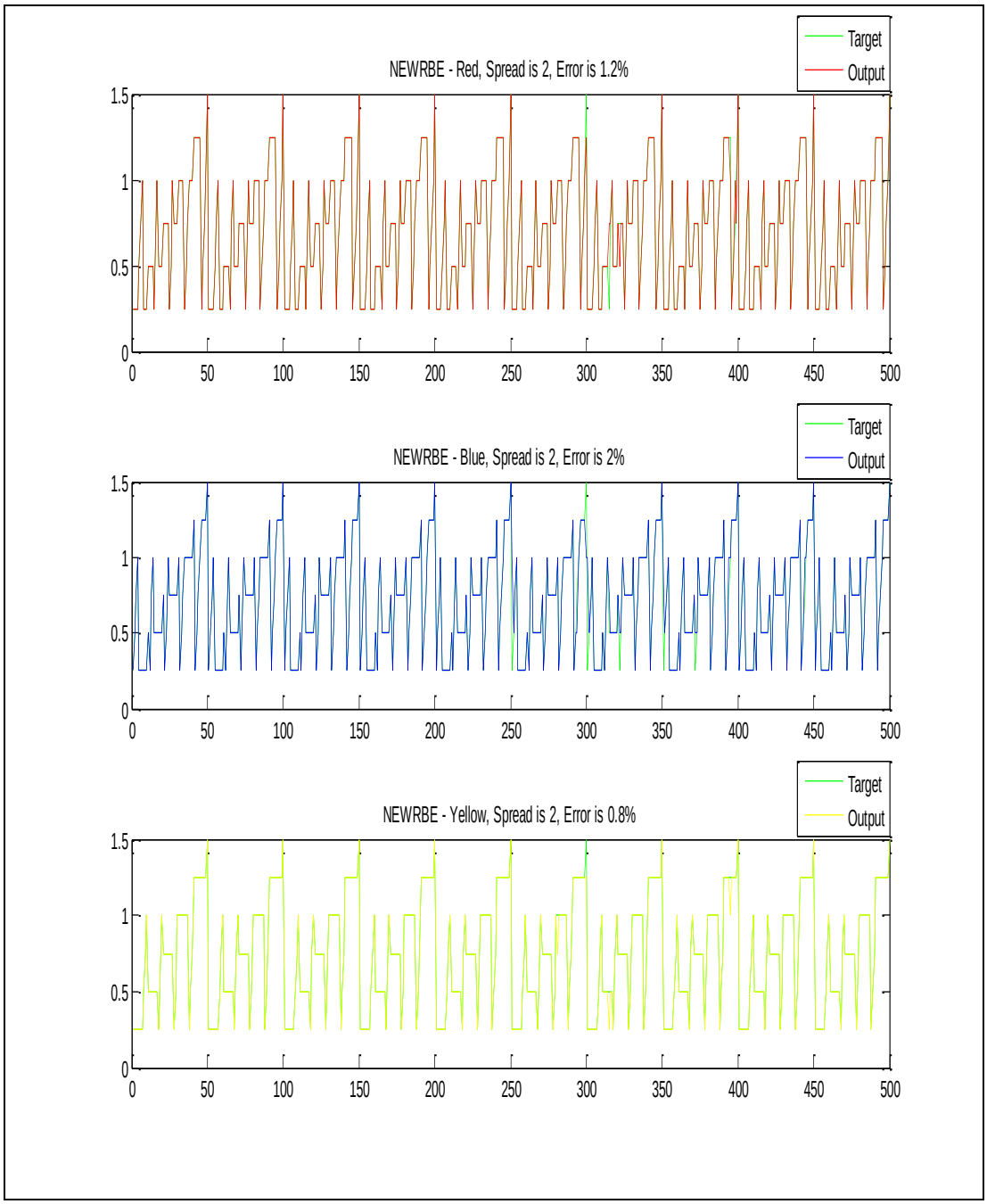

Figure 5.17 The output distribution after rounding outputs to a limited range

In this operation rounding was done to the output values. By this operation, it was seen that the error percentages were decreased in a very acceptable range. The percentages are about 0-2 according to the colors as seen in Figure 5.17.

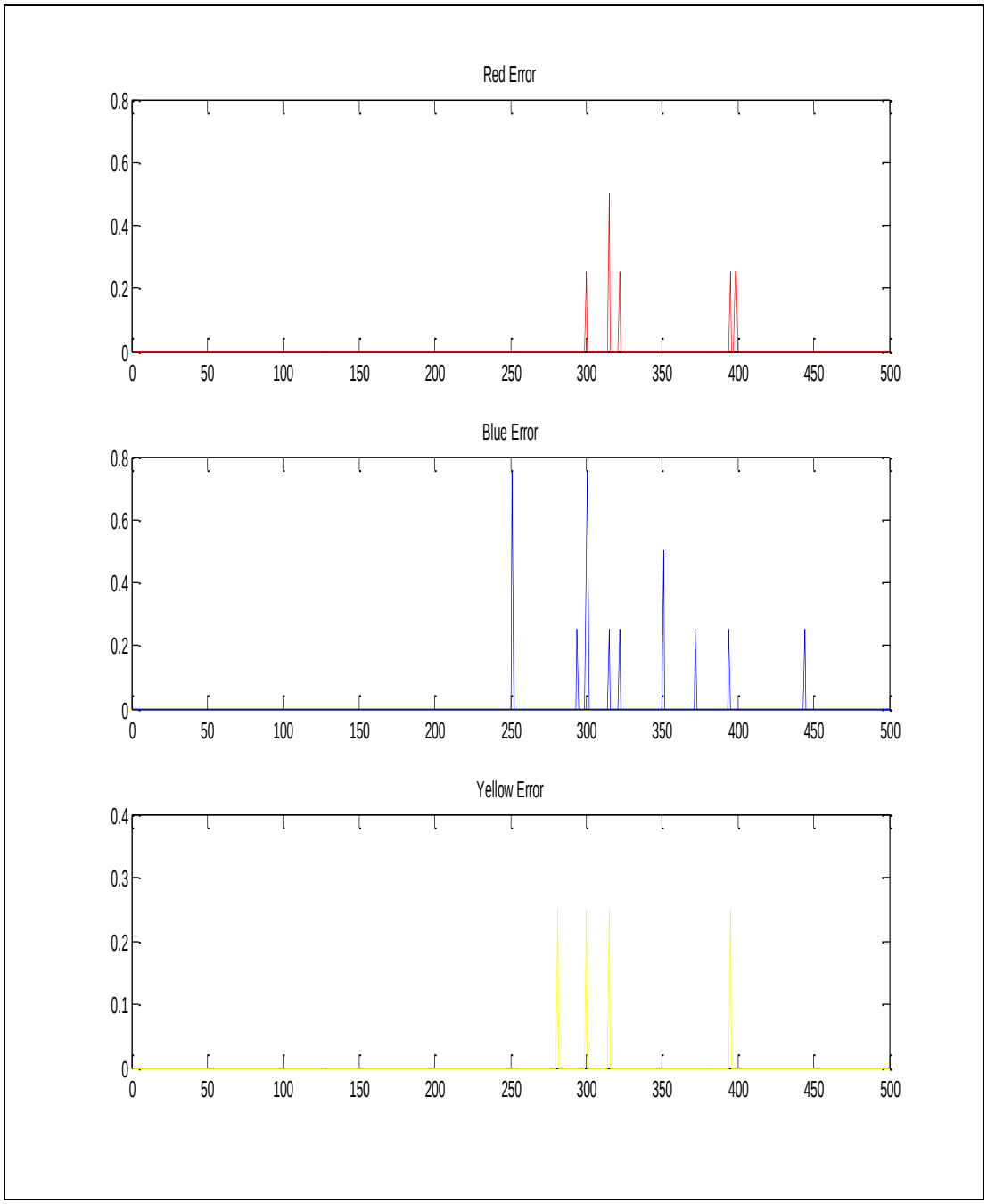

Figure 5.18 The error values after rounding outputs to a limited range

After the operation of rounding, it was observed that the amplitudes of the error values were decreased. Also, it was seen from the Figure 5.18 that, the quantity of the error values were decreased in an acceptable range.

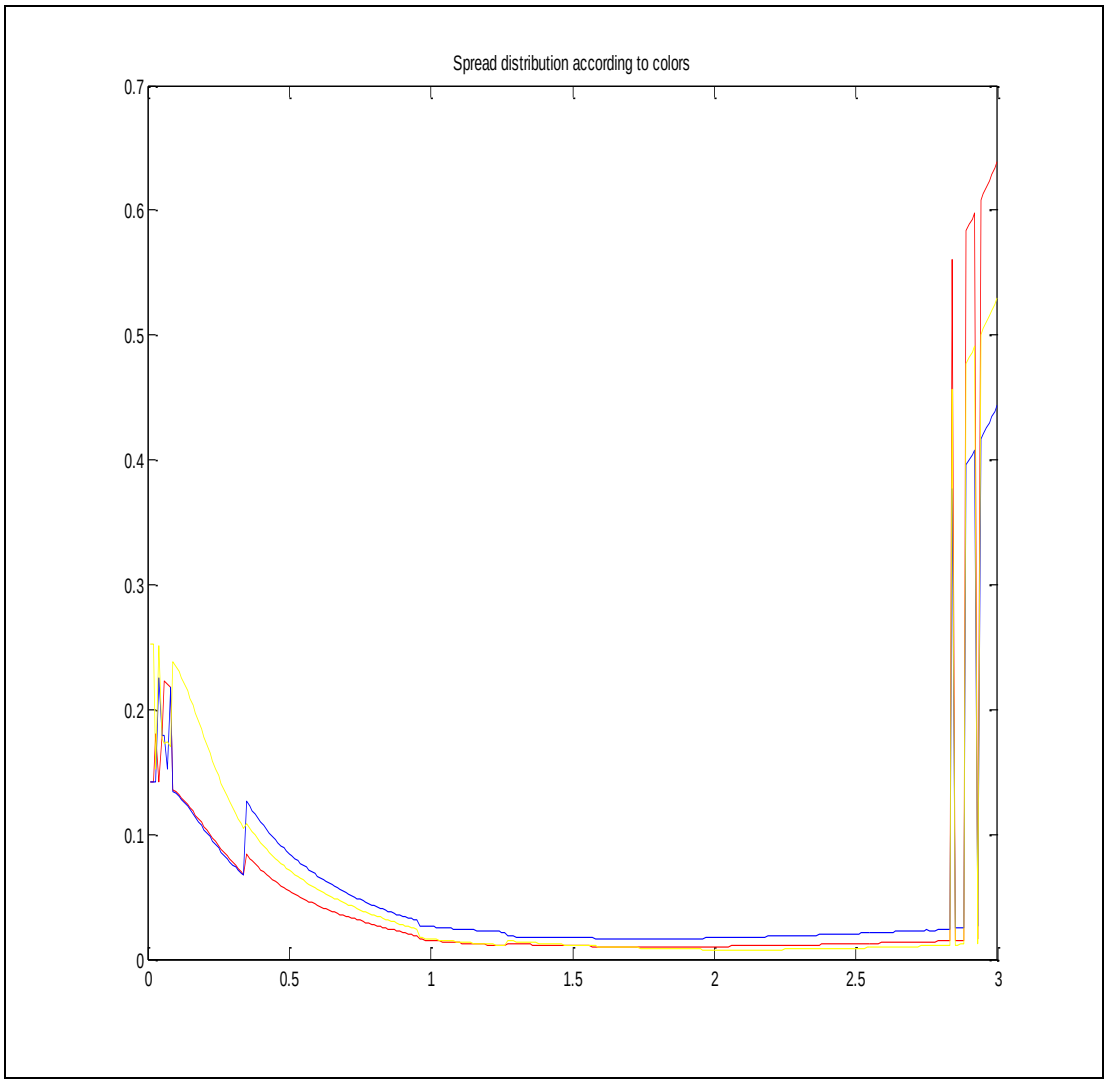

Figure 5.19 Spread distribution according to colors in the range 0-3.

According to Figure 5.19, the spread values between 0-3 are given with respect to the error values.

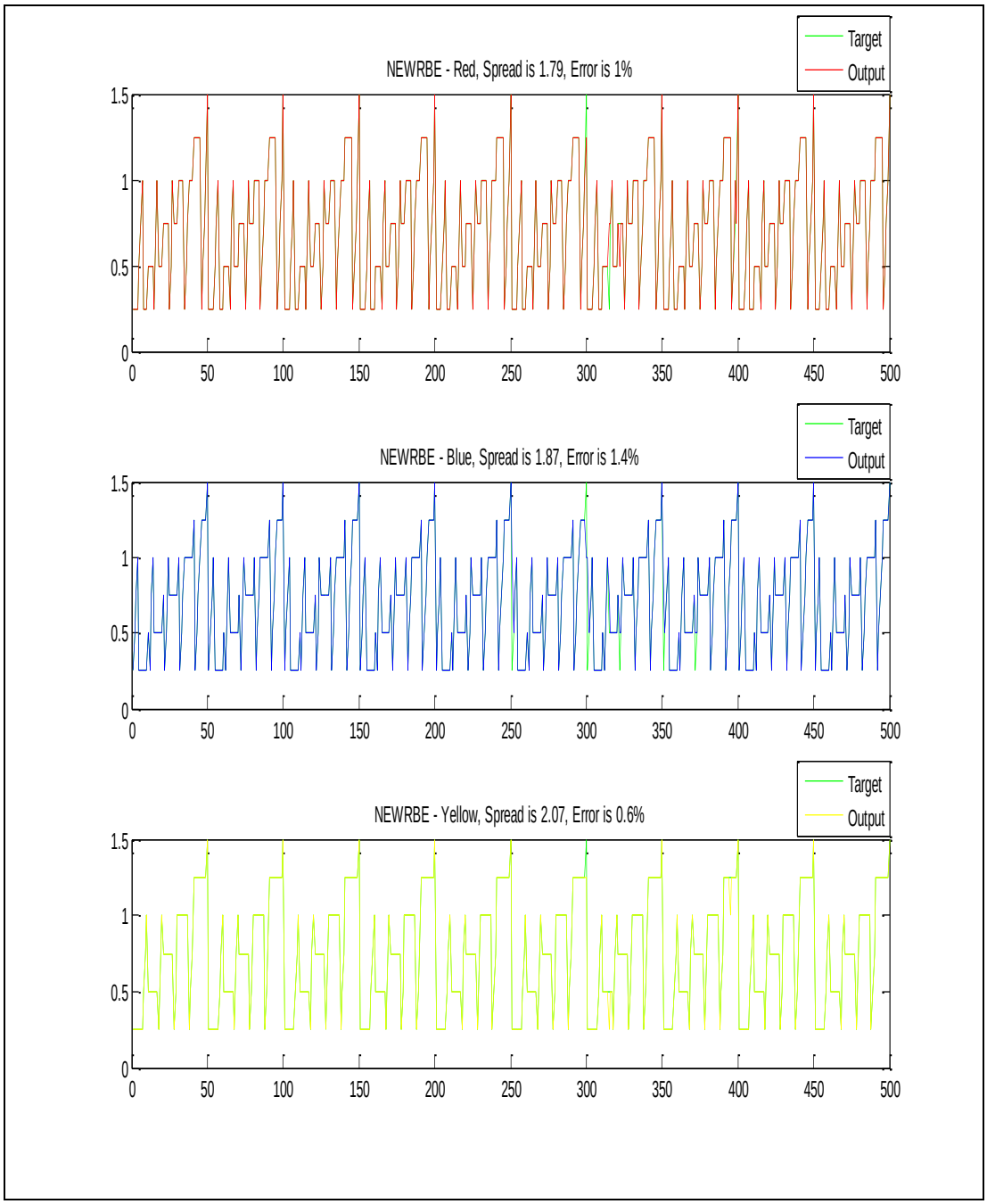

Figure 5.20 The output distribution after rounding outputs to a limited range and zooming the spread value to the range of 0-3

It was recognized that in the previous trials the spread values were always between 0 and 3. In the previous trials, the interval of spread was 1 when the range of spread was 1-100. In this trial it was decided to observe the range of 0-3 spread values detailed and decreasing the intervals of spread distribution from 1 to 0.01. It was seen that, calculating the spread value more detailed made the system more successful. It is seen in Figure 5.20.

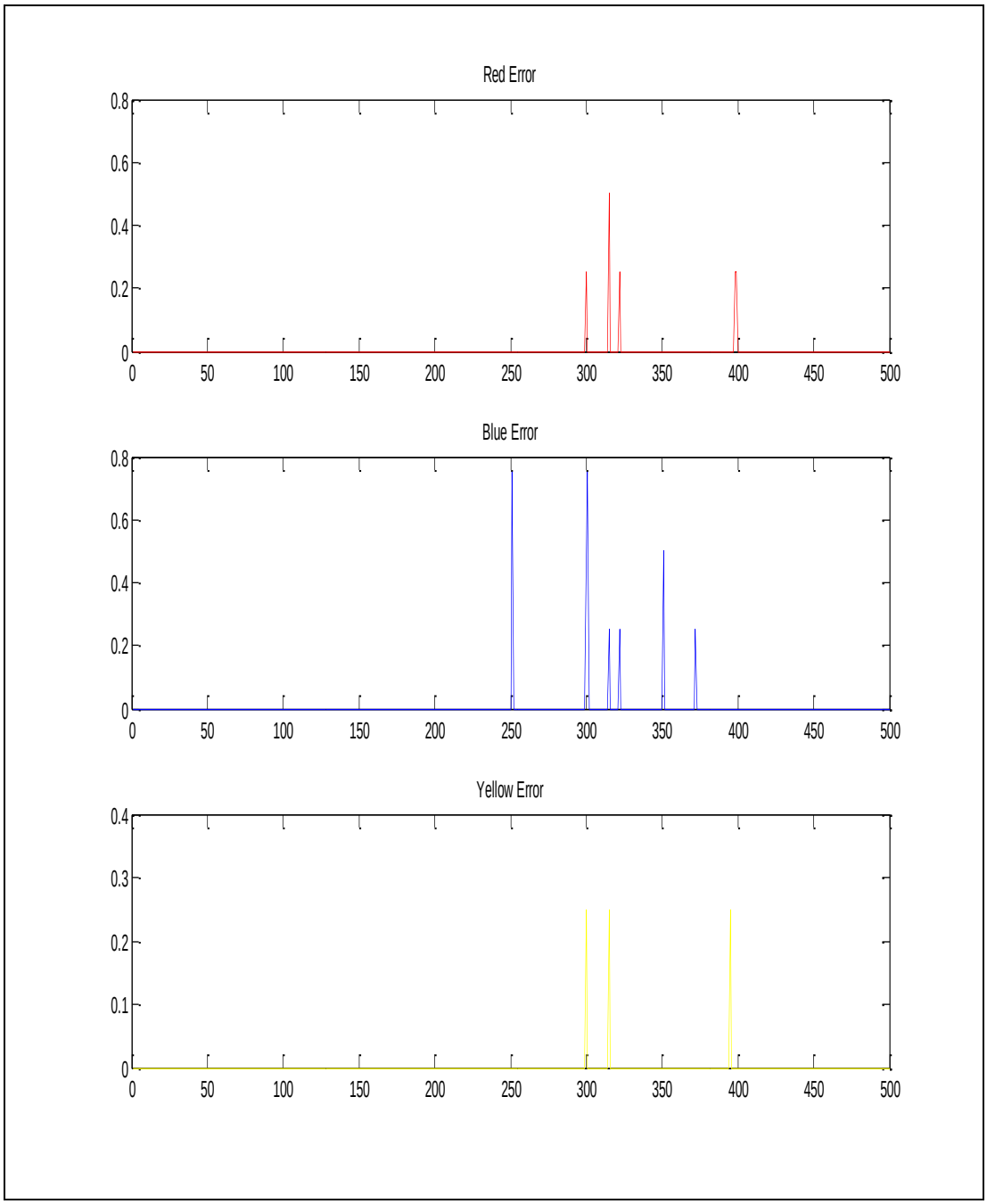

Figure 5.21 The error values after changing the spread distribution to the range of 0-3

It was observed in the Figure 5.21 that the error quantity was decreased when the spread distribution was decreased to a range of 0-3.

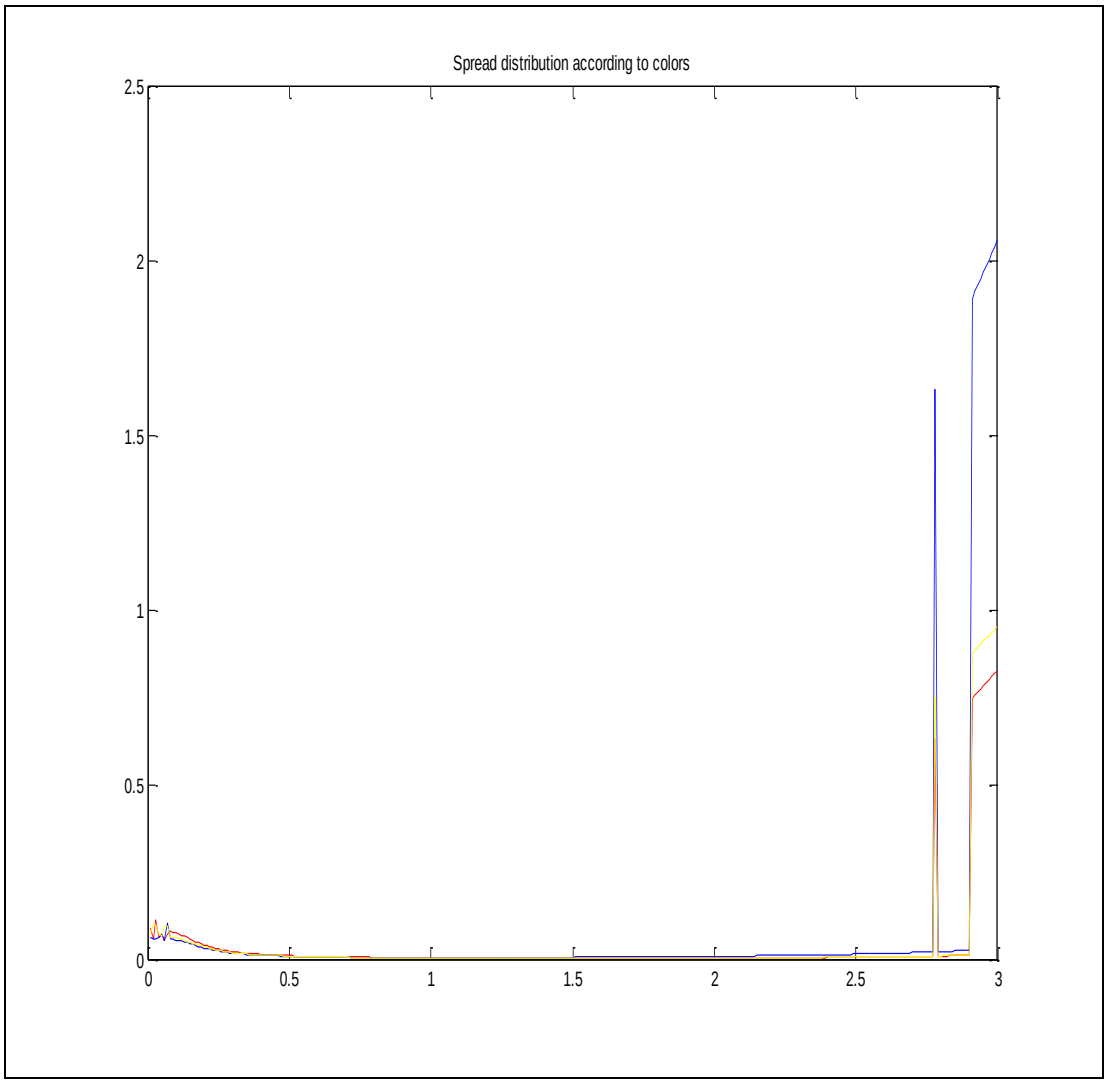

Figure 5.22 Spread distribution according to colors

According to Figure 5.22, the spread values between 0-3 are given with respect to the error values.

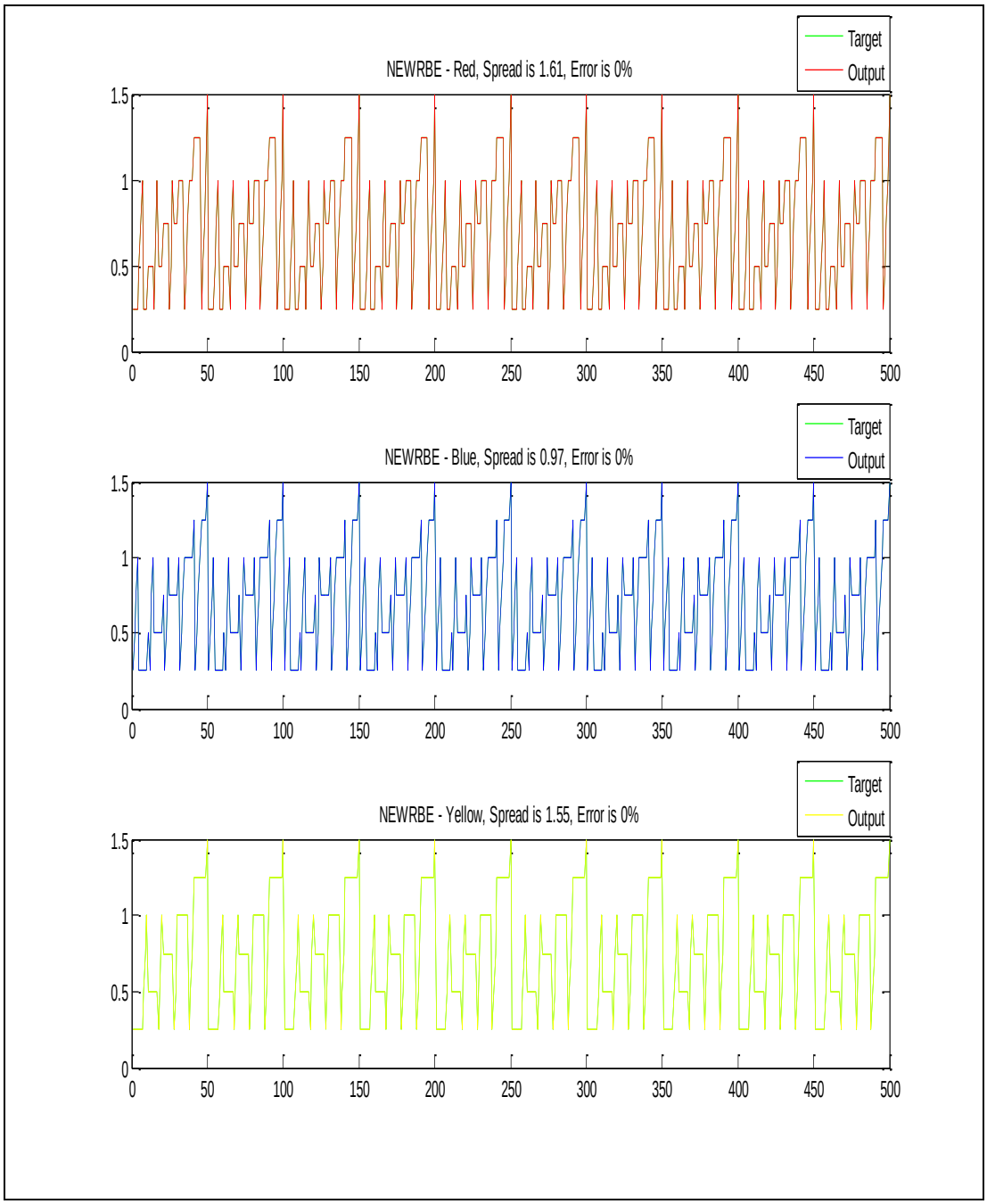

Figure 5.23 The output distribution after rounding outputs to a limited range, zooming the spread value and increasing the training samples

It was seen in the previous applications that the number of training samples was not enough. By the way, in this application it was decided to increase the number of training samples from 250 to 400. This increasing means that the number of repeats of dyeings was increased from 5 to 8. As it can be seen from Figure 5.23, the error percentages of all of the three colors were decreased to 0%. This means a success of 100%.

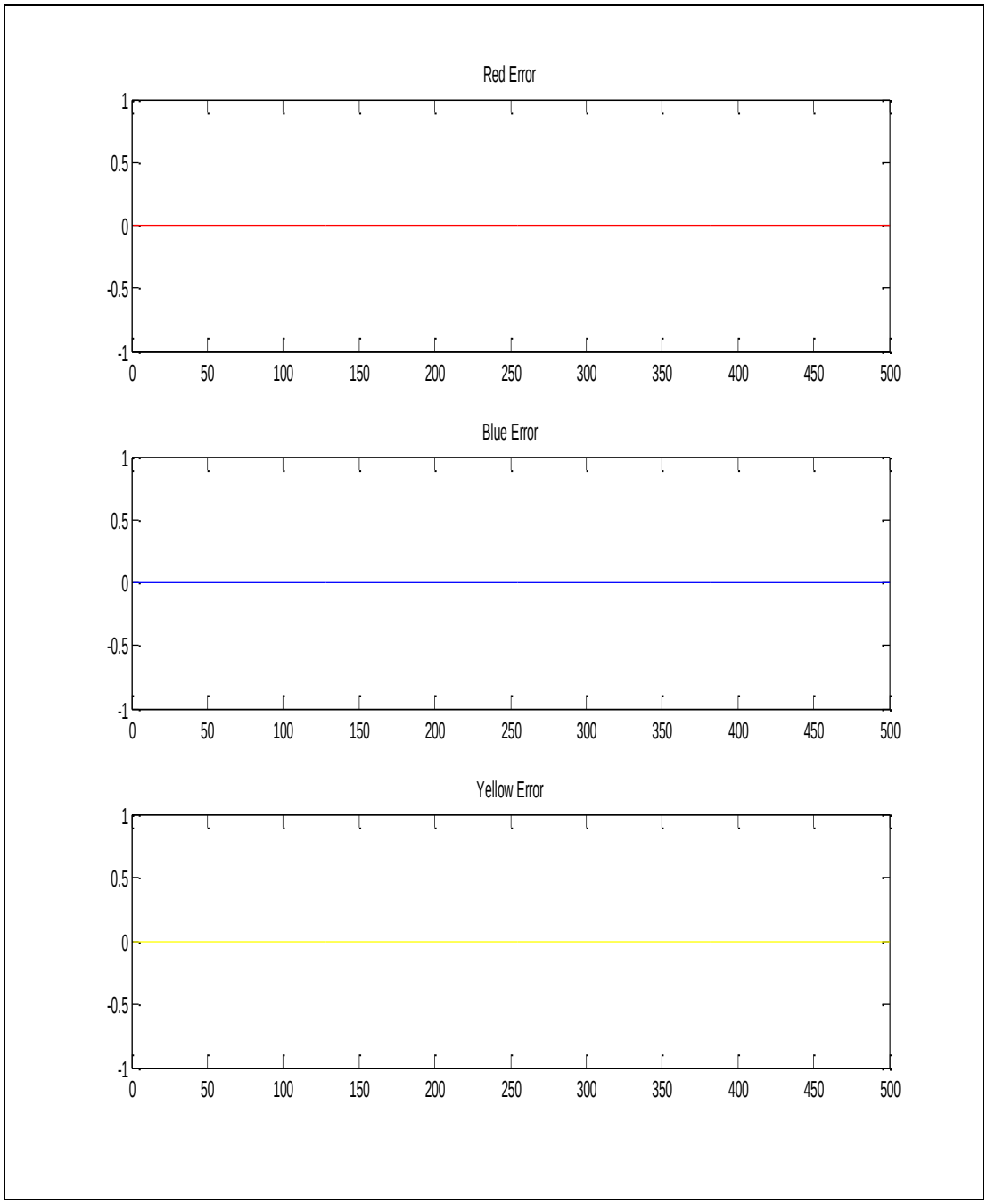

Figure 5.24 The error values after increasing the number of training samples

As it can be seen from Figure 5.24, all of the magnitudes and quantities of error values for three outputs were decreased to 0.

While implementing newrb, we did the same changes explained in the newrbe. Besides, in newrb a parameter called goal was used. By changing this parameter it is decided when to stop the training process.

5.2.2 MLP Applications

In Matlab, there are three functions related to MLP. These functions are newff, newcf, newelm. Applications were done with all of them. After observing the results, one of them was found more successful comparing to the others. This method is newff. In the rest of the application, only this method was used and its results are given in the result part.

While applying newff, we did the following changes and explained as below.

As it is known from the theoretical part, in MLP there are some hidden layers between the input and the output layer. We changed the number of hidden layers and the neuron numbers in the hidden layers. It was observed that the performance of the net decreased after increasing the number of hidden layers more than two. In the same way, increasing the hidden neuron number too much decreased the performance. Moreover, increasing the hidden layer neuron number and the number of the hidden layers increased the training time enormously. Because of this it was not an efficient way to increase the hidden layer neuron number and hidden layer number, too much. As a result, in this part the number of hidden layers was chosen as one or two.

The resultant graphs and explanations of these graphs are given below for this application.

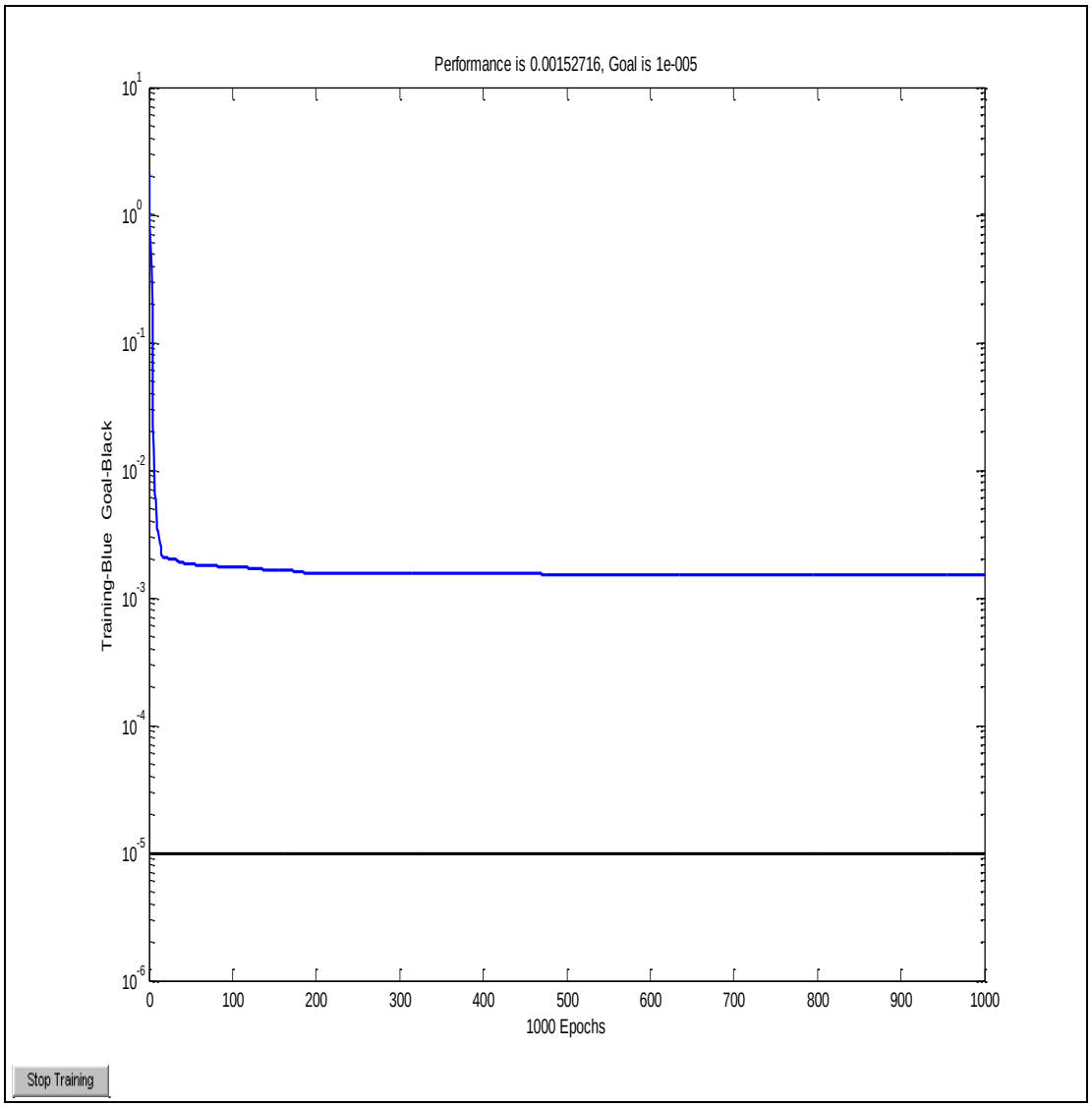

Figure 5.25 The training performance of 1 hidden layer and 10 hidden neurons

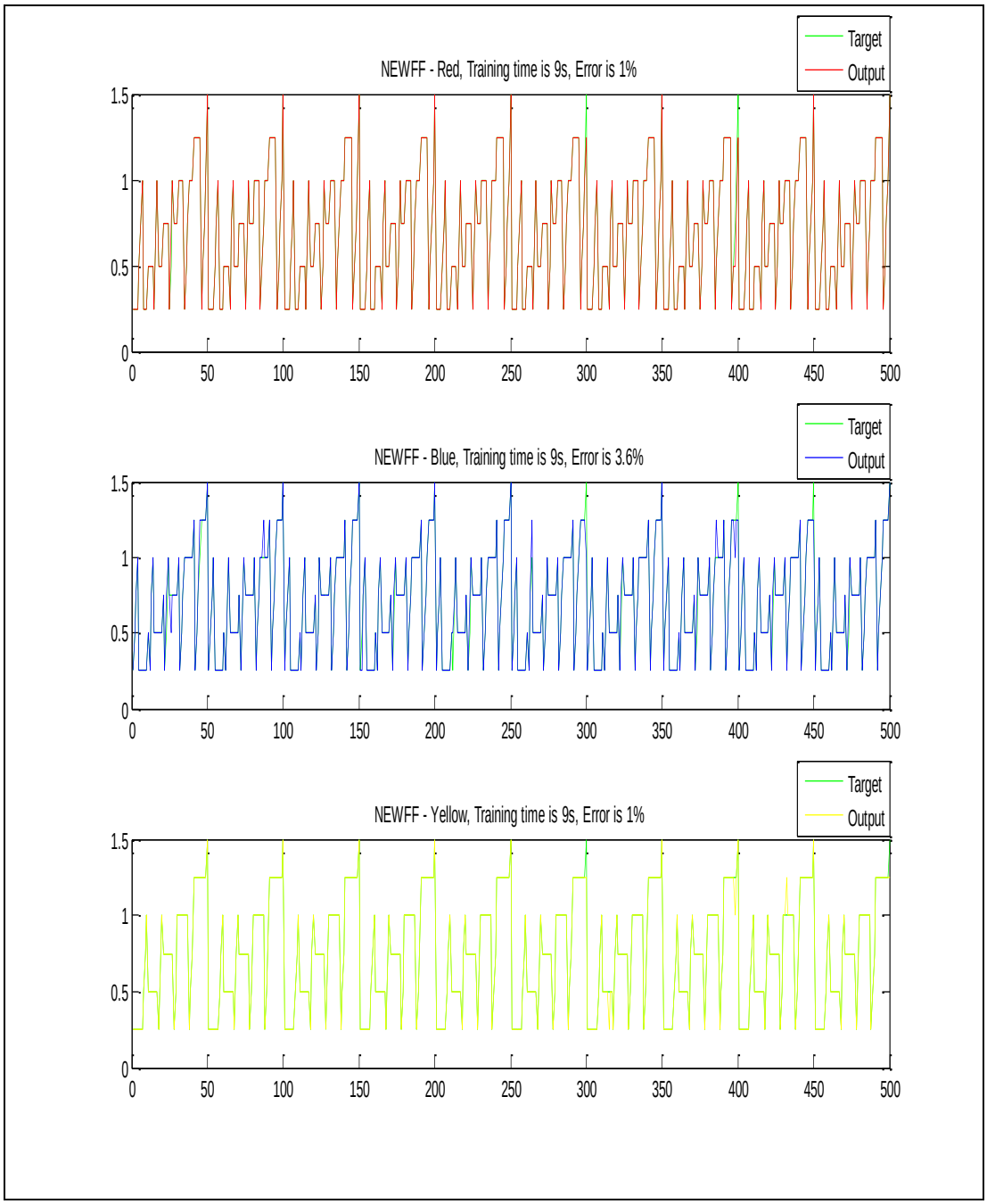

Figure 5.26 The output distribution after training with 1 hidden layer and 10 hidden neurons

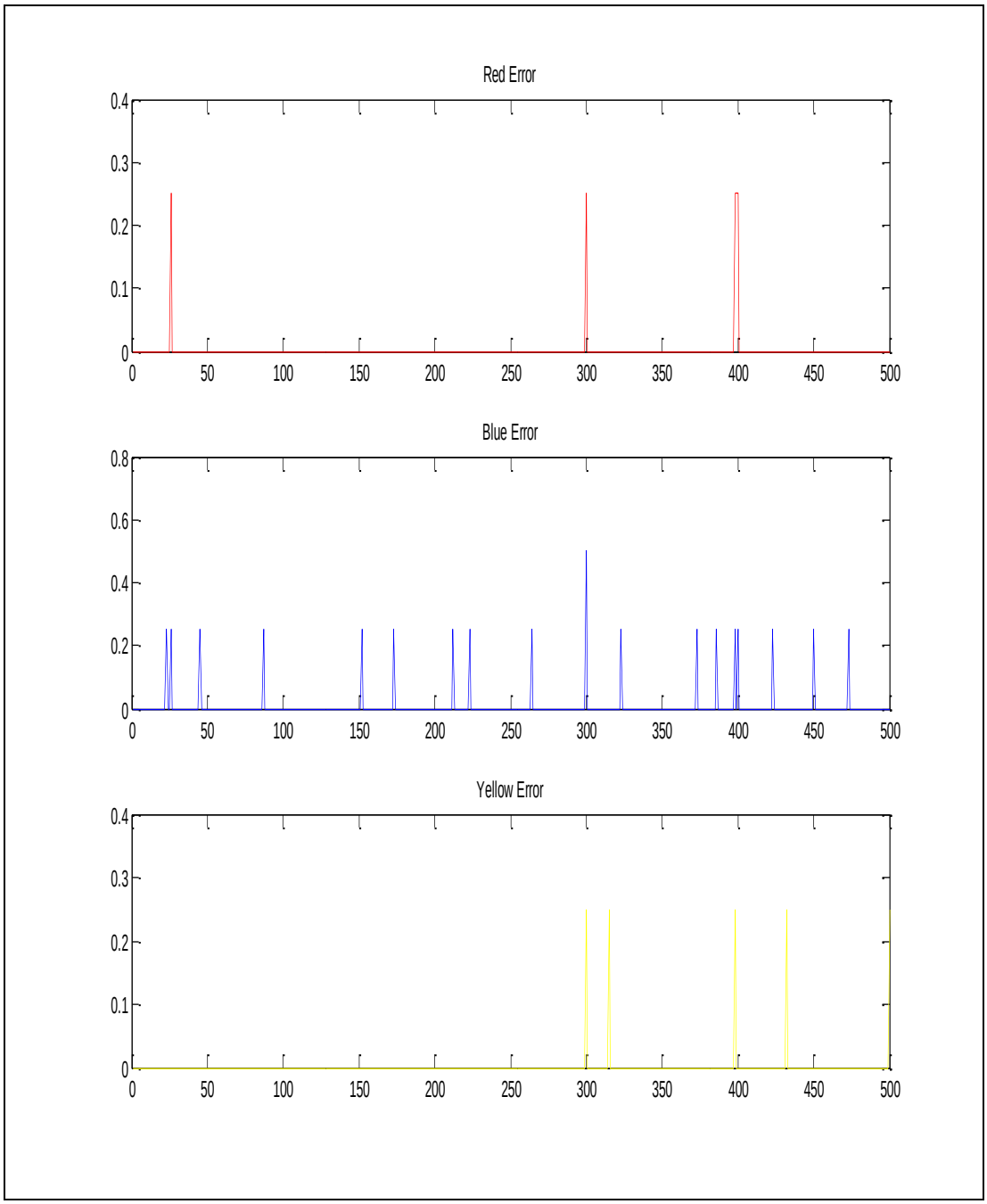

Figure 5.27 The error values of 1 hidden layer and 10 hidden neuron system

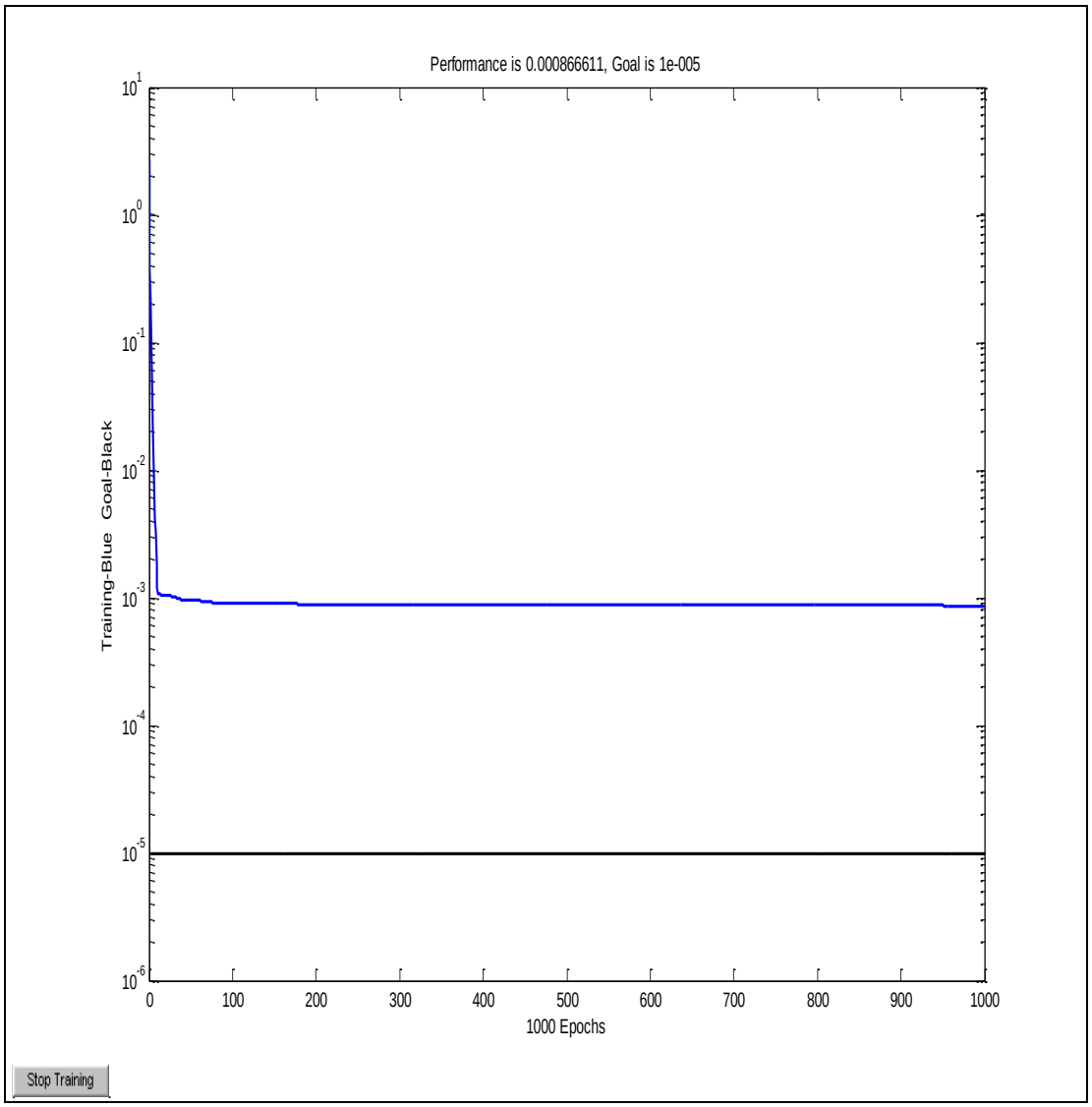

Figure 5.28 The training performance of 1 hidden layer and 25 hidden neurons

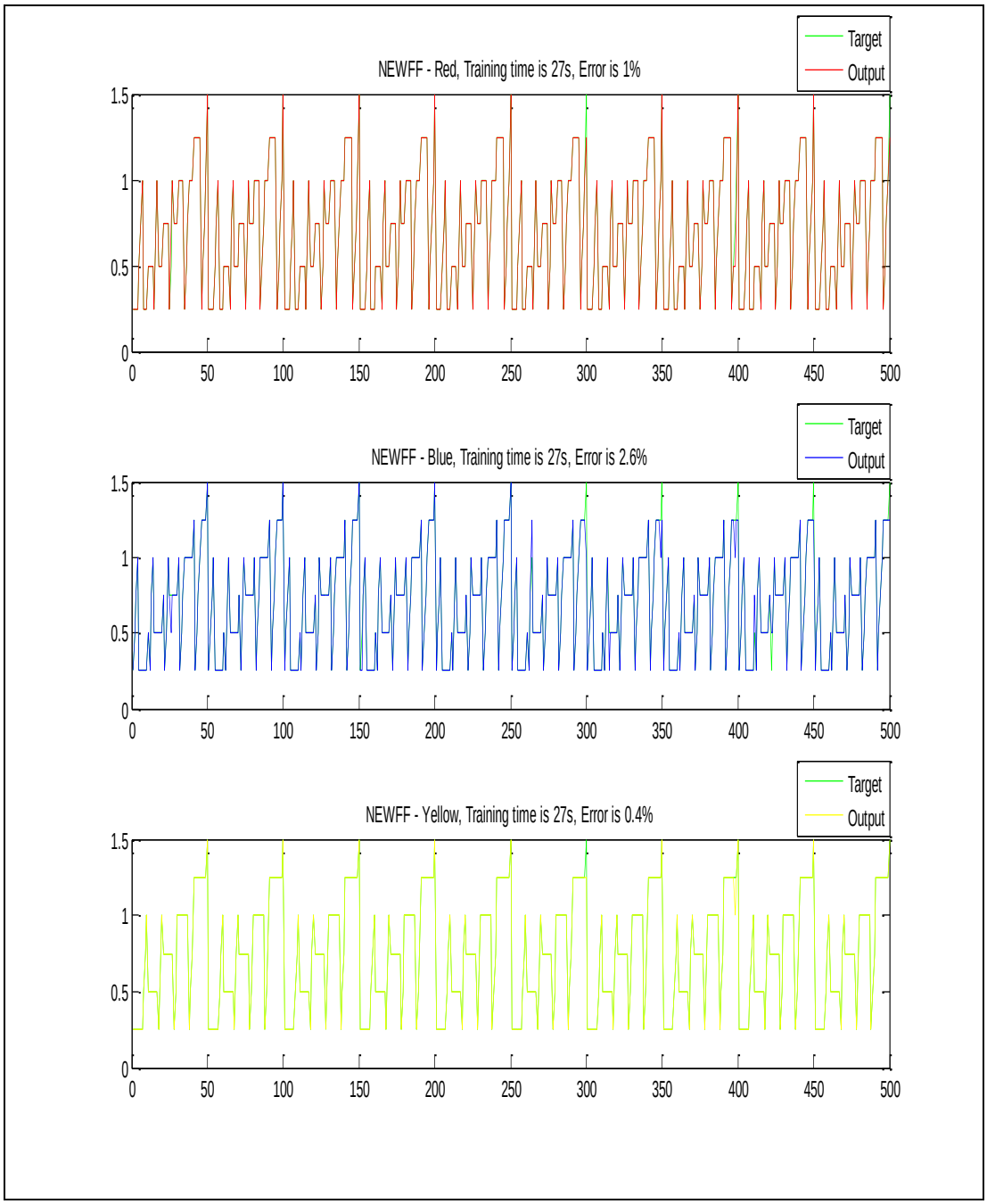

Figure 5.29 The output distribution after training with 1 hidden layer and 25 hidden neurons

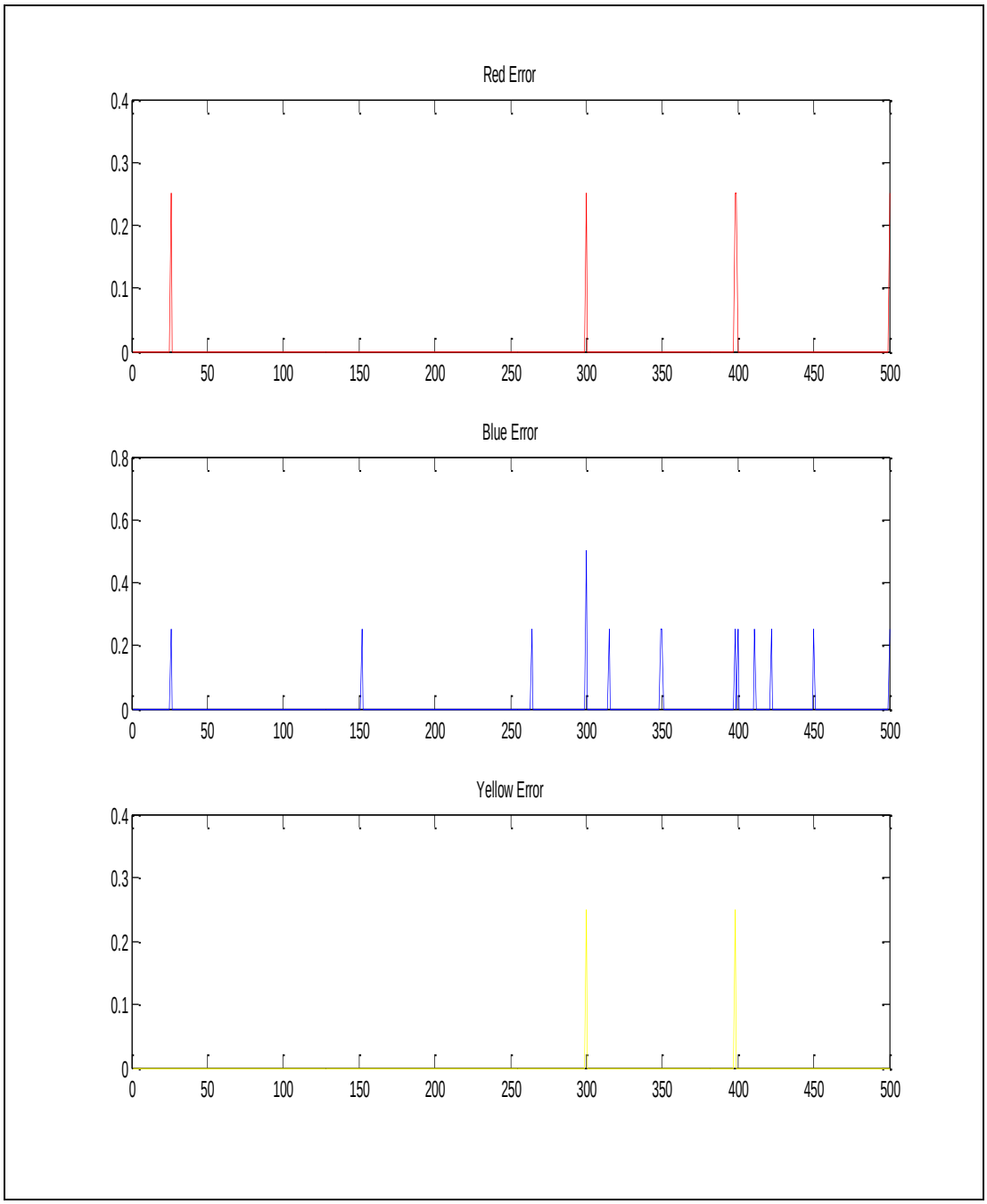

Figure 5.30 The error values of 1 hidden layer and 25 hidden neuron system

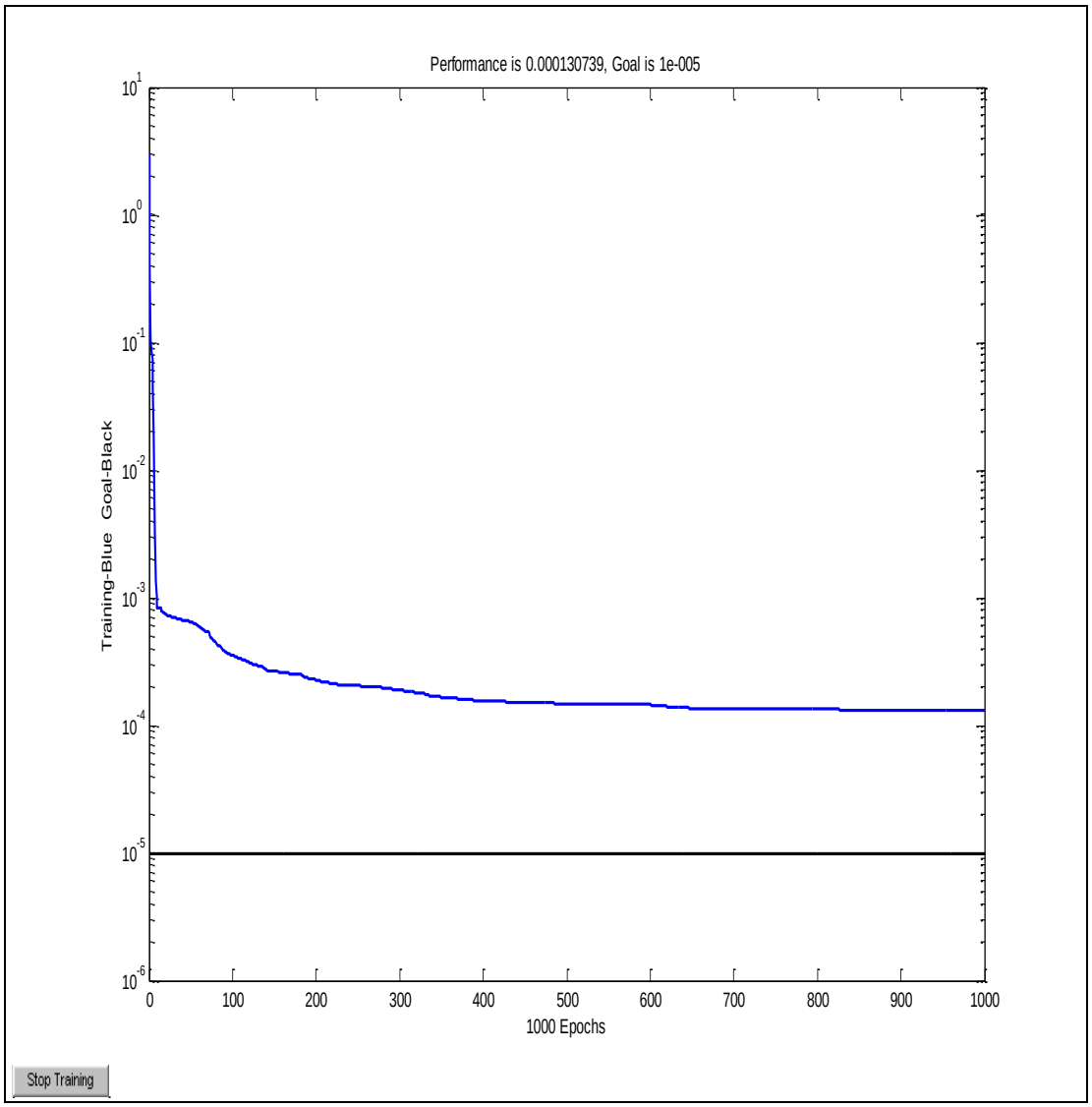

Figure 5.31 The training performance of 1 hidden layer and 50 hidden neurons

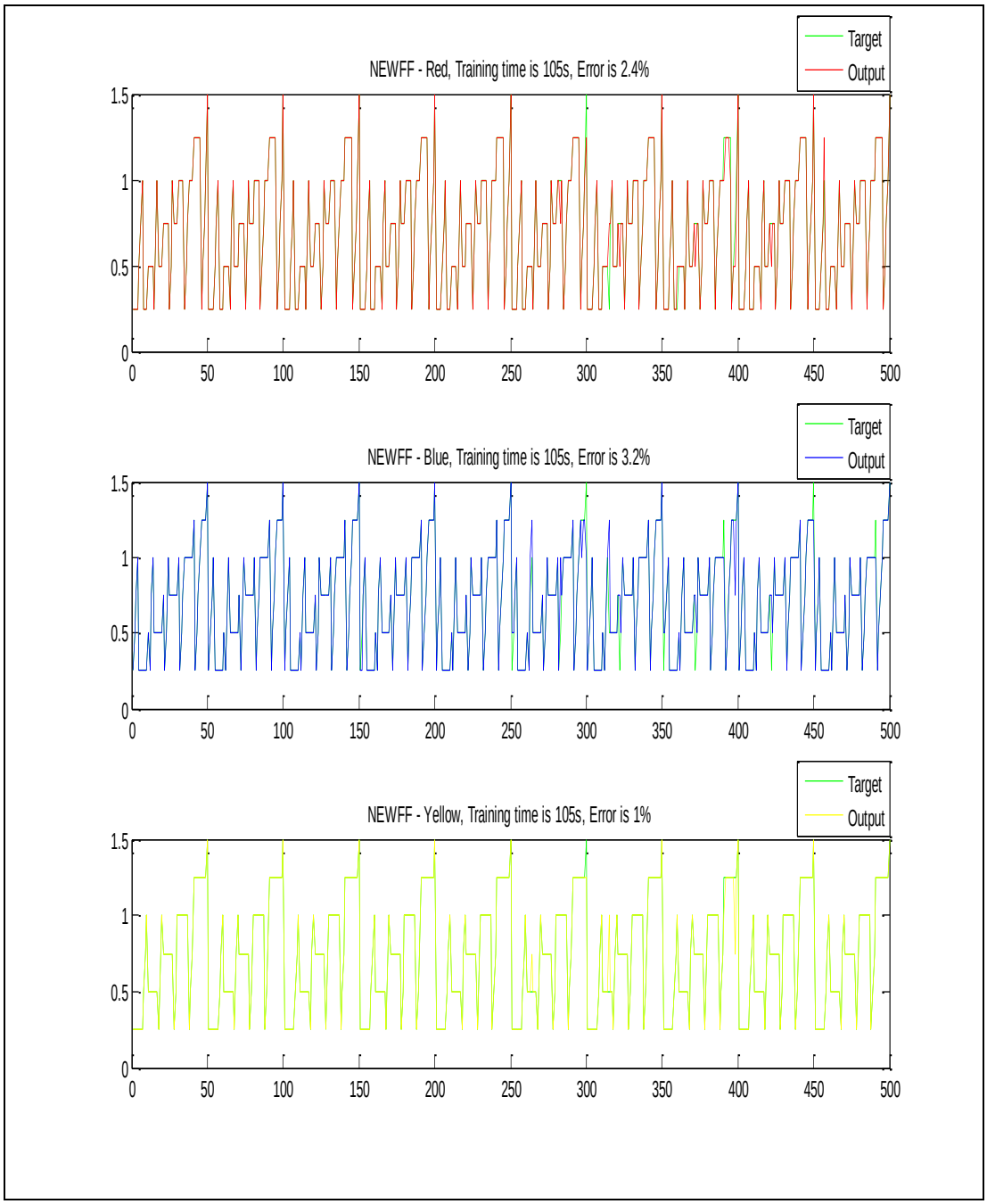

Figure 5.32 The output distribution after training with 1 hidden layer and 50 hidden neurons

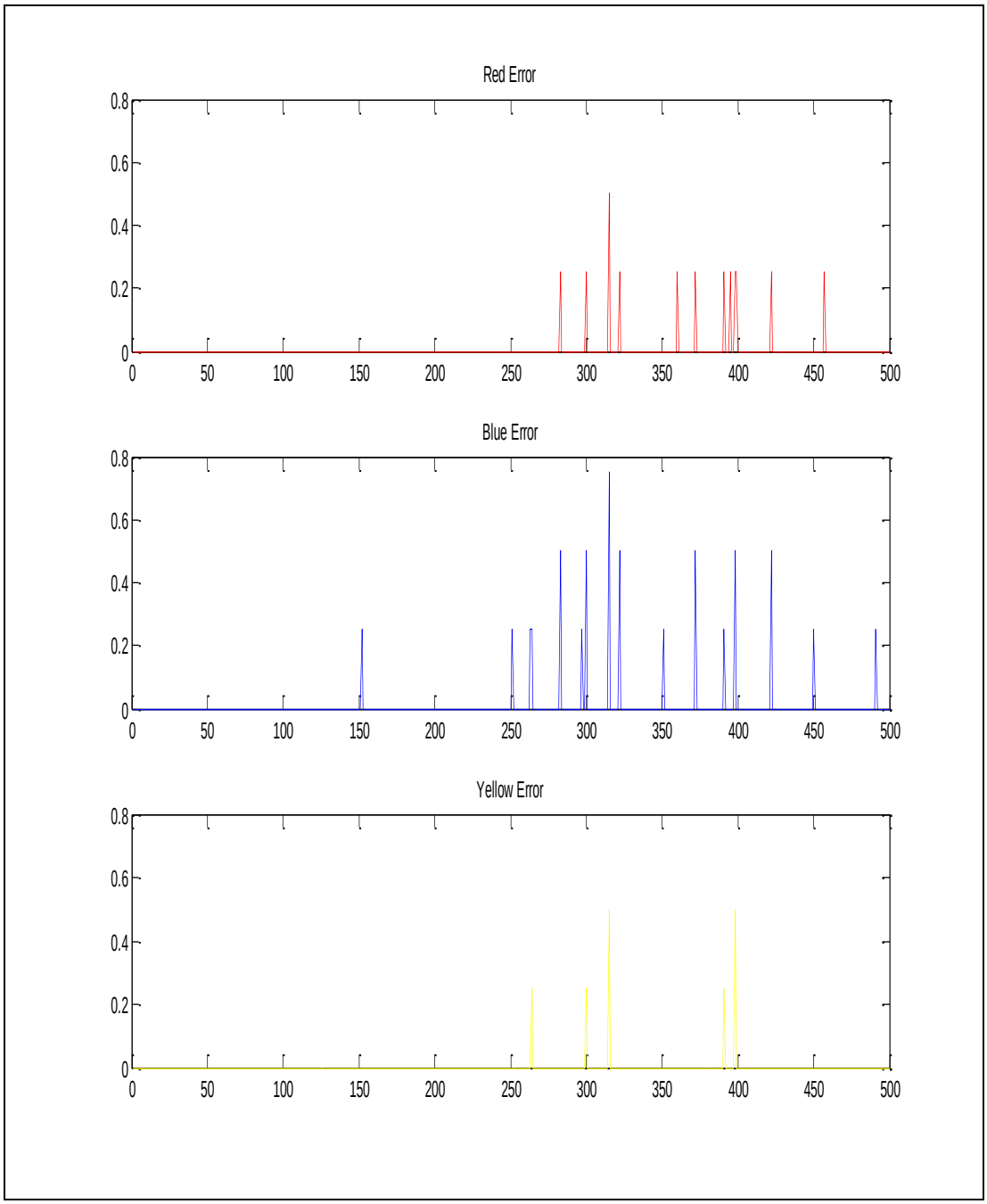

Figure 5.33 The error values of 1 hidden layer 50 hidden neuron system

In the training first of all, the hidden neuron number was increased from 10 to 25. It was seen that the time of training was increased from 9 sec to 27 sec as seen in Figures 5.26 and 5.29. However, the training performance was increased according to Figure 5.25 and Figure 5.28, and error percentages were decreased from the Figure 5.27 and Figure 5.30.

Secondly, it was tried to observe that what will happen if the hidden neuron number increases from 25 to 50. It was seen that both the training time and error percentages increased according to Figures 5.29, 5.32 and Figures 5.30, 5.33 respectively. However the training performance of the system was decreased as shown in Figures 5.28 and 5.31.

These experiments show that increasing the neuron number too much does not mean that the system will have a good performance of training.

In Matlab, while applying MLP, as mentioned before we used newff command. One of the parameters that affect the usage of this command is the selection of the transfer function. The transfer function can be any differentiable transfer function such as tansig, logsig, or purelin. All of them were applied to the system. Finally it was decided to use in the output layer the transfer function of purelin, in the hidden layer the transfer function of logsig. These transfer functions were more successful compared to the others.

The other parameter that affects the training of the data is the epoch number. The epoch number specifies when the training stops. If it reaches the epoch number we decided before, it stops training. The epoch number was changed so many times and it is seen that if it is too large, the training does not converge to a result in an acceptable time. If it is too small, then, the training process fails.

The Figures from 5.36 to 5.42 and explanations of these figures are given below for this application.

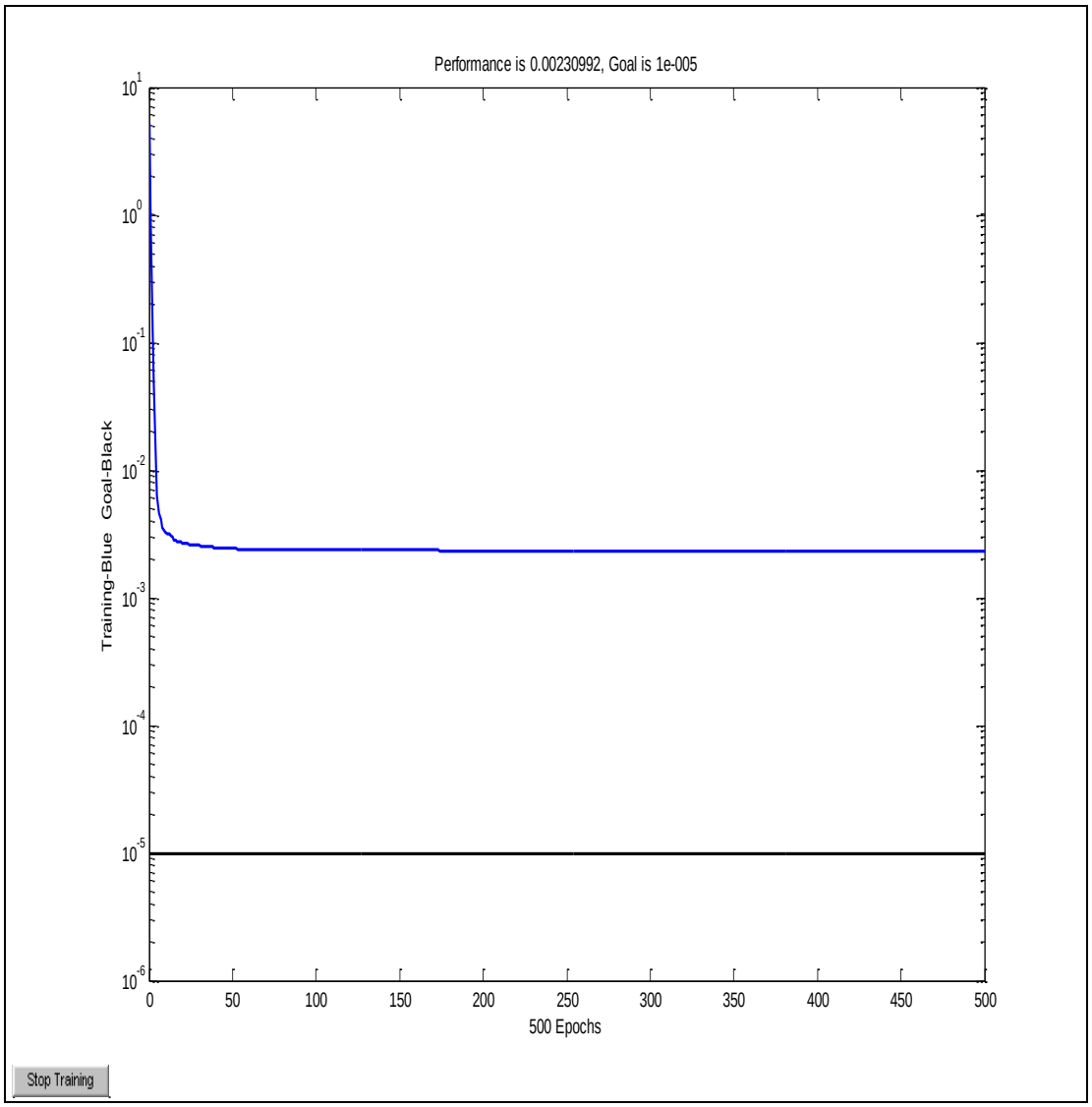

Figure 5.34 The training performance of 500 epochs

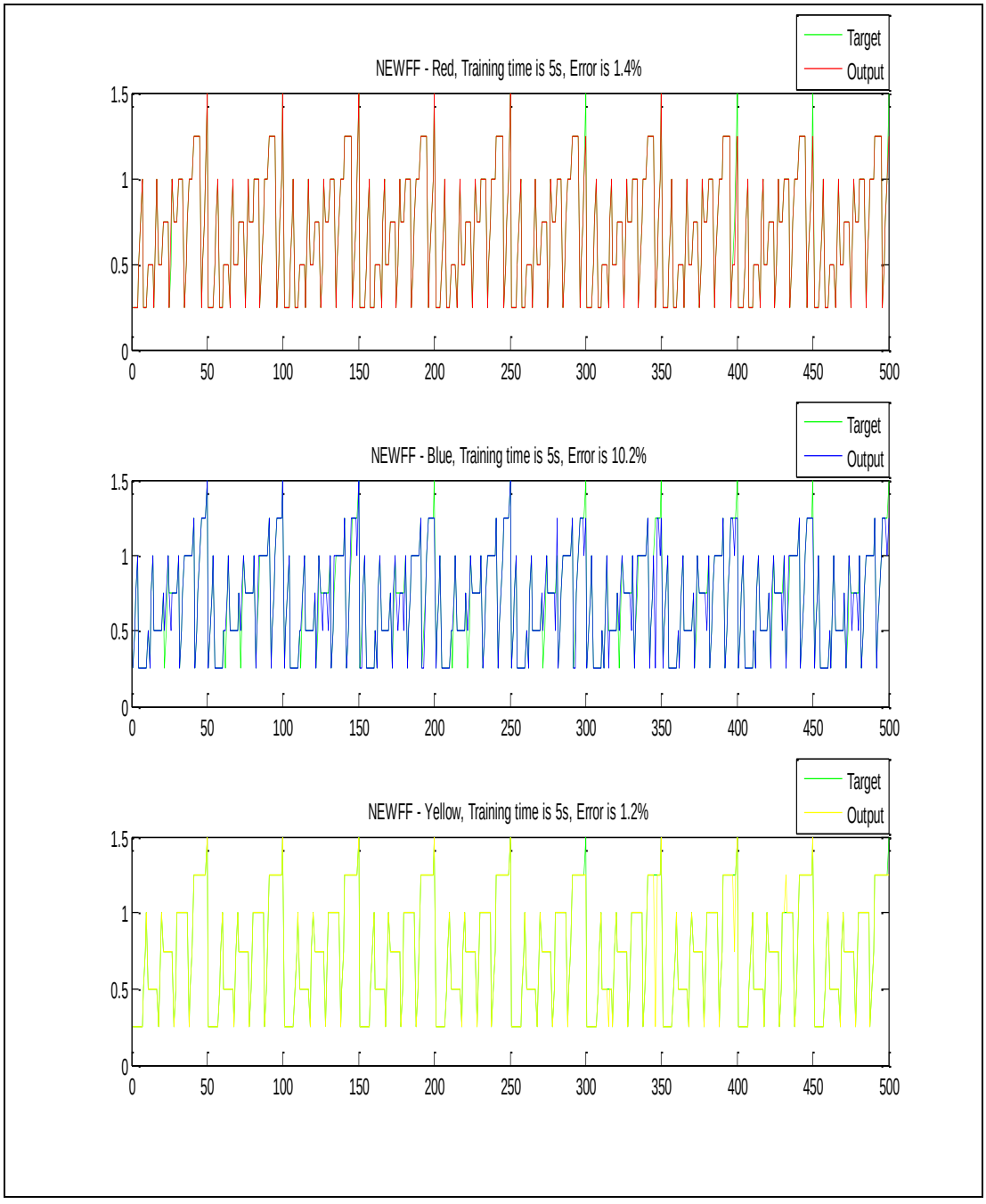

Figure 5.35 The output distribution after training with 500 epochs

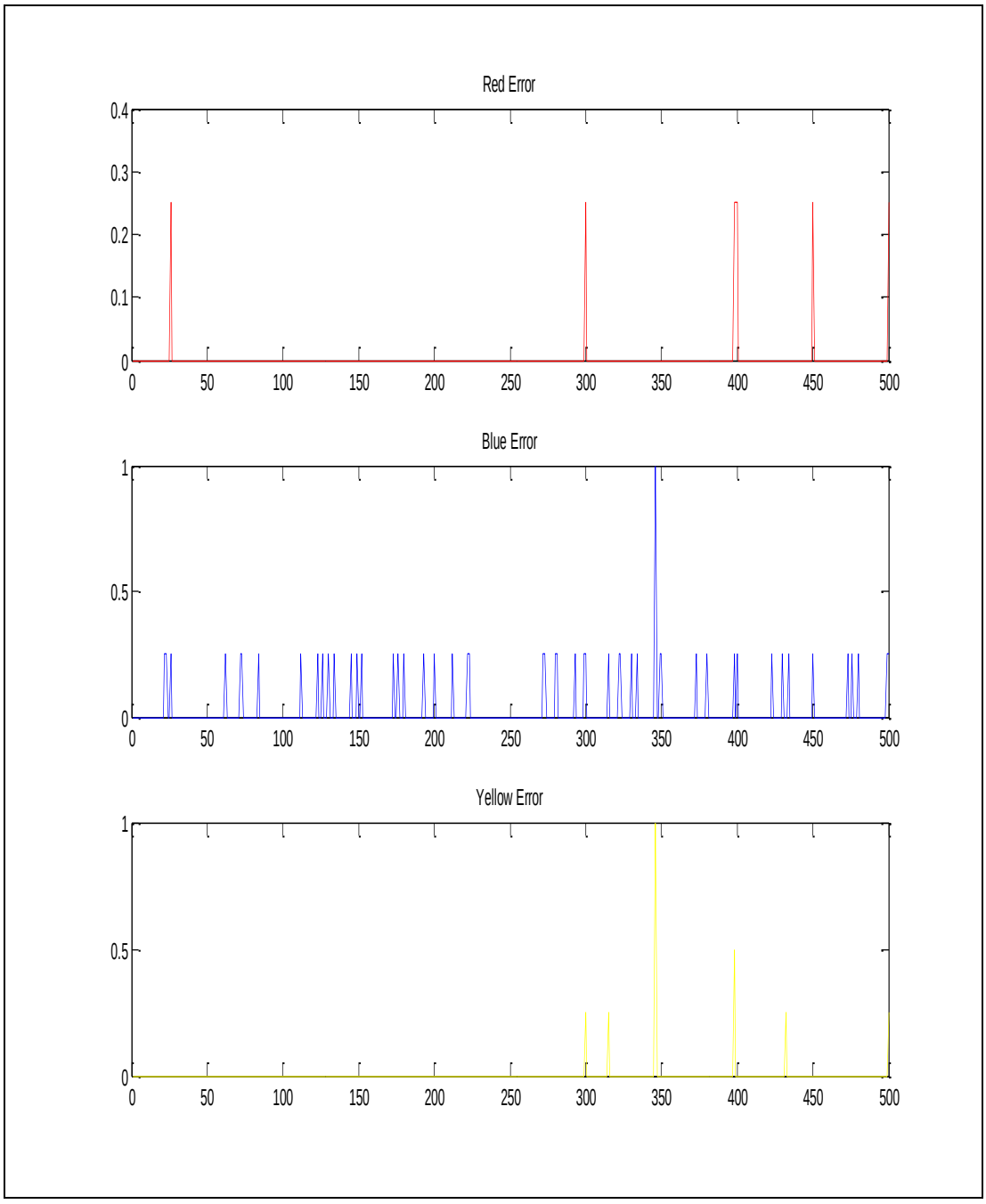

Figure 5.36 The error values of training with 500 epochs

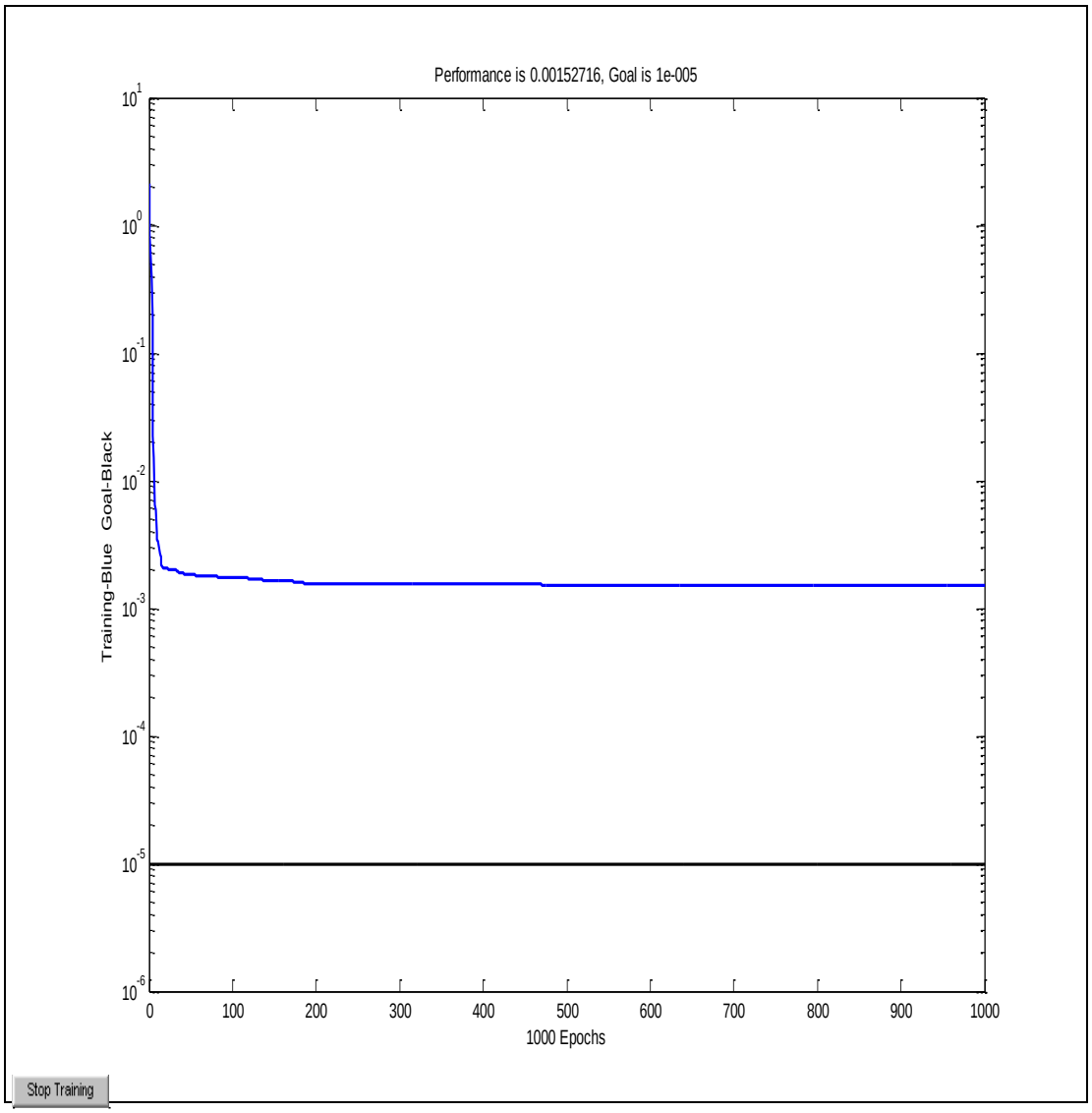

Figure 5.37 The training performance of 1000 epochs

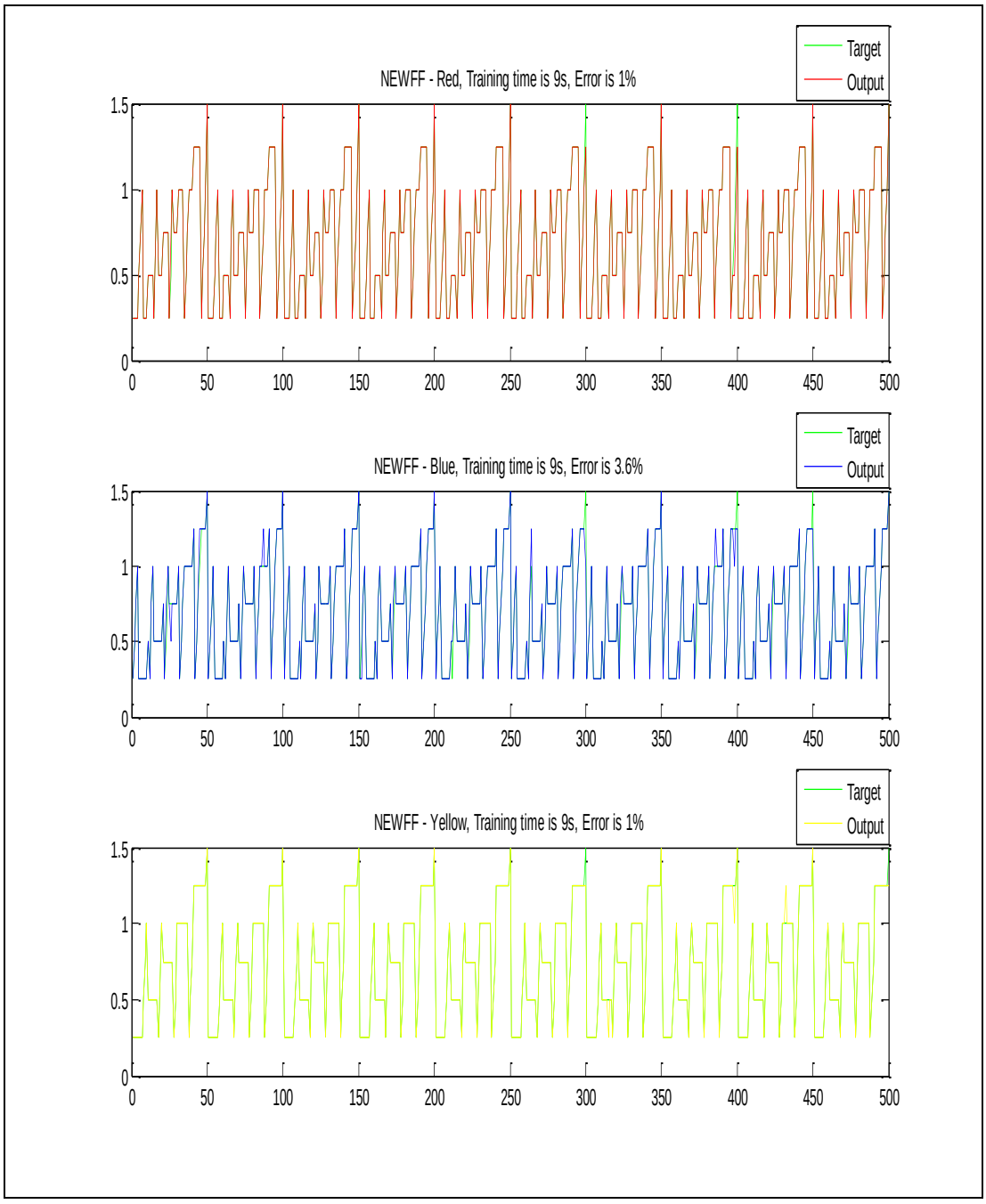

Figure 5.38 The output distribution after training with 1000 epochs

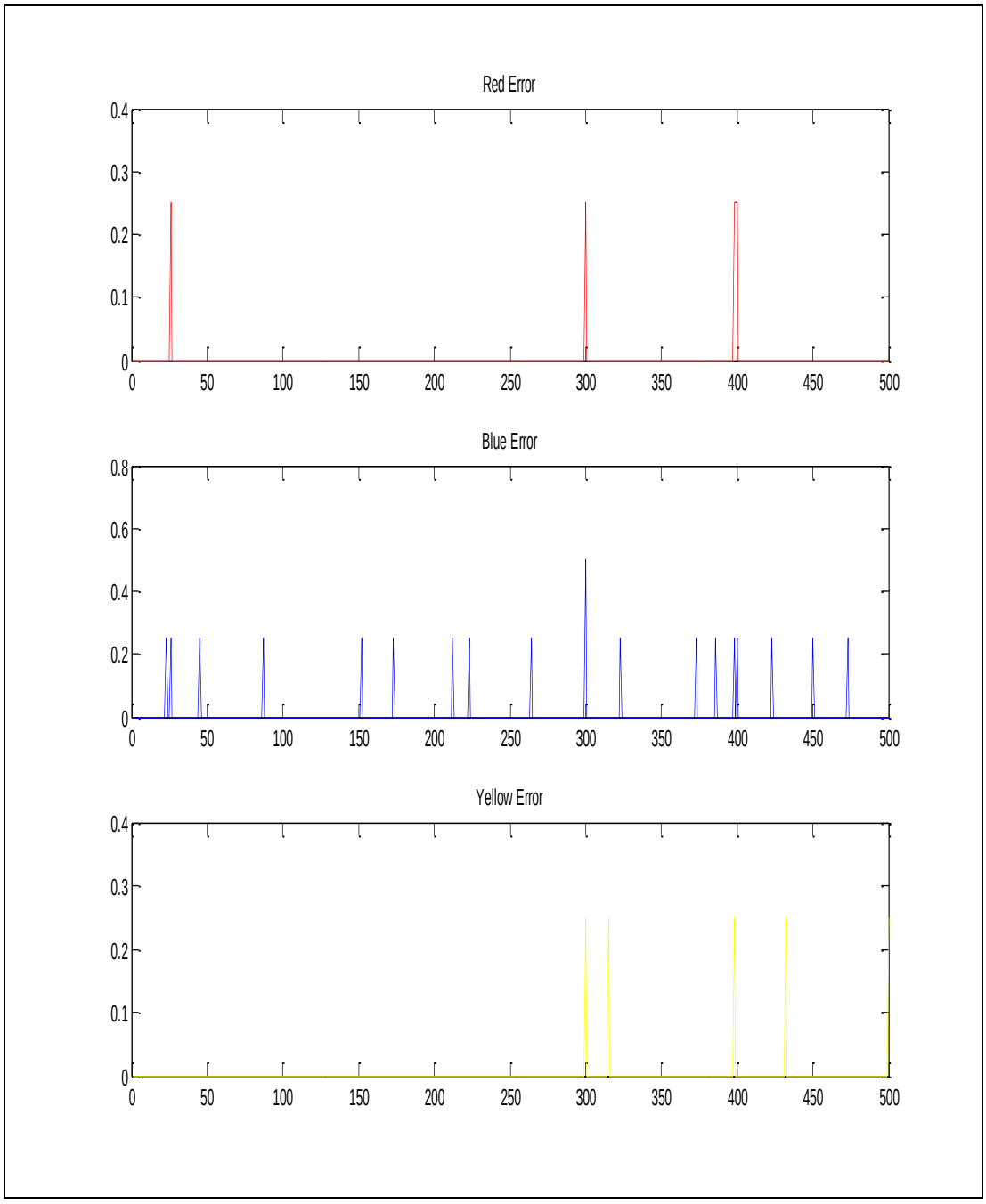

Figure 5.39 The error values of training with 1000 epochs

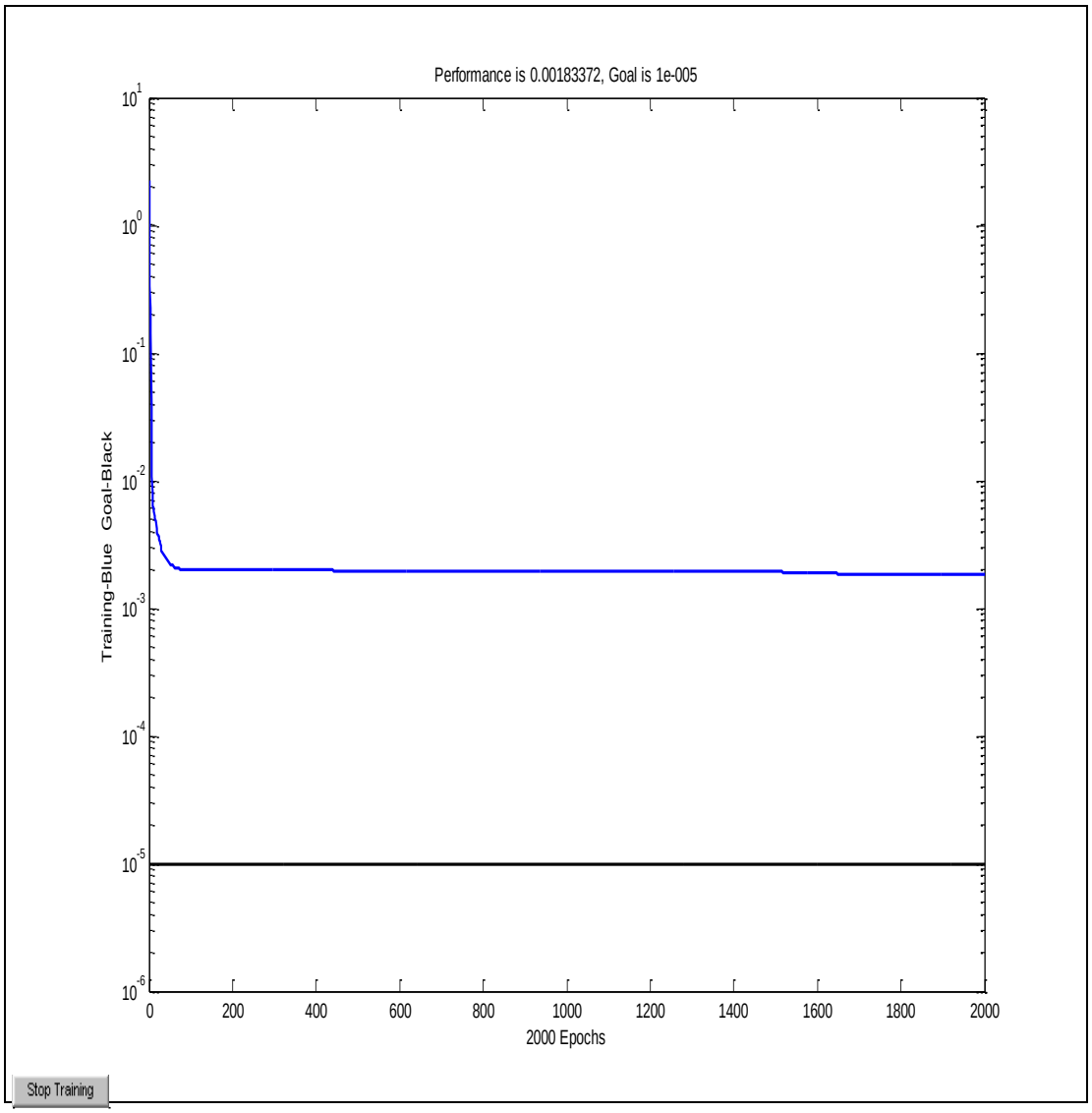

Figure 5.40 The training performance of 2000 epochs

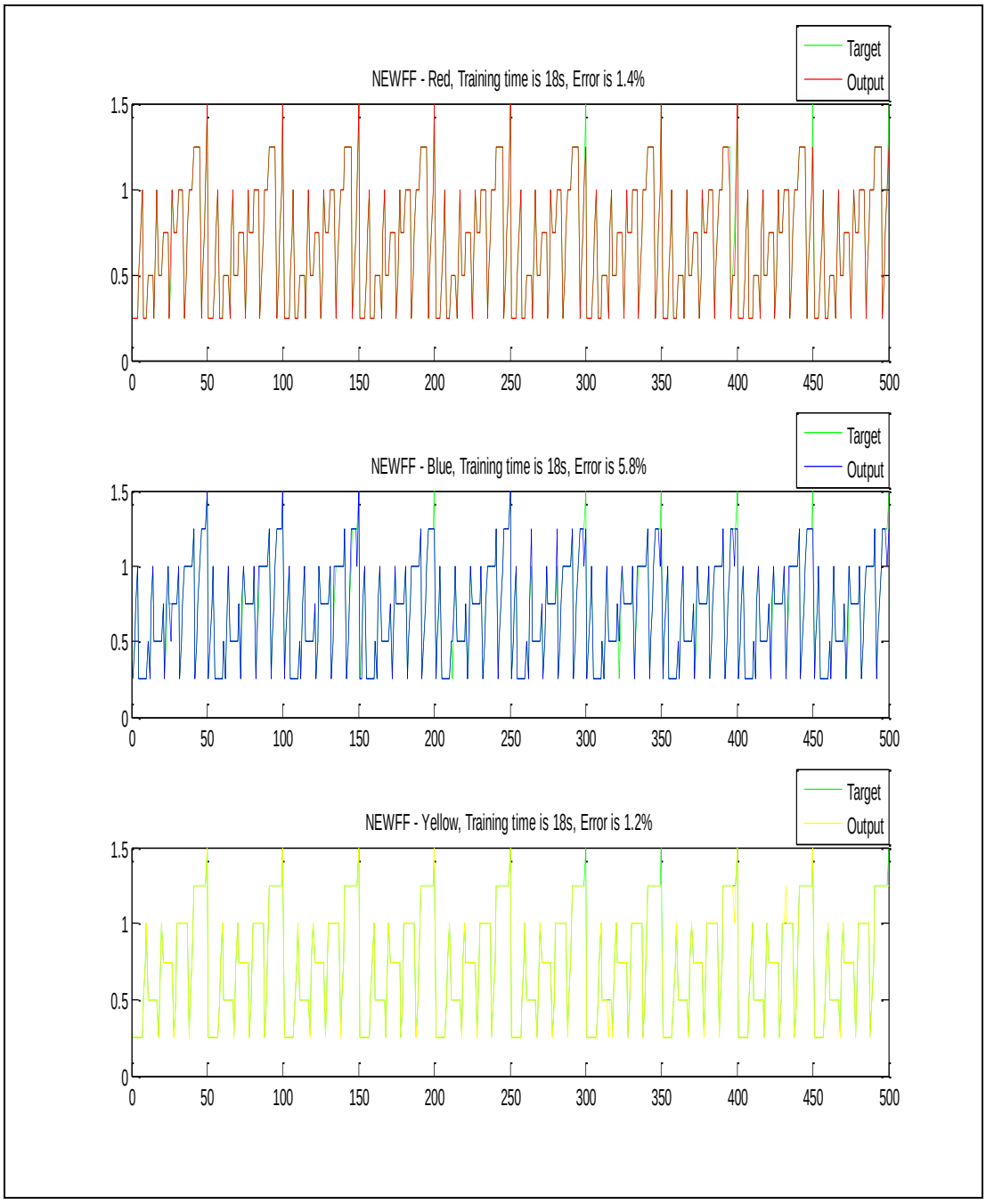

Figure 5.41 The output distribution after training with 2000 epochs

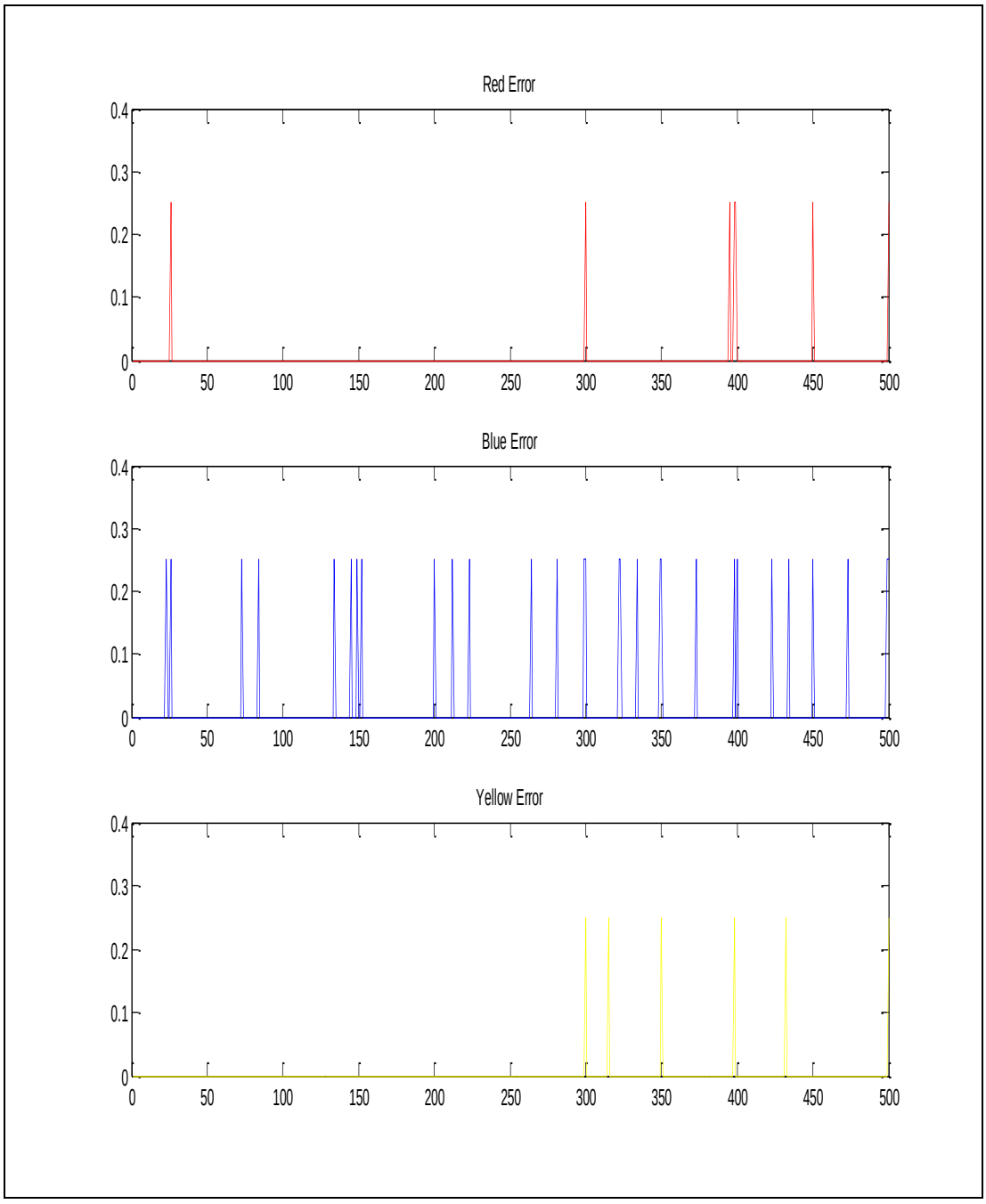

Figure 5.42 The error values of training with 2000 epochs

In the training first of all, the epoch number was increased from 500 to 1000. It was seen that the time of training was increased from 5 sec to 9 sec. However, the training performance was increased and error percentages were decreased.

Secondly, it was tried to observe that what will happen if the epoch number increases from 1000 to 2000. It was seen that both the training time and error

percentages increased. However the training performance of the system was decreased.

These experiments show that increasing the epoch number too much does not mean that the system will have a good performance of training. After some time, the system was overtraining. This means, the system loses the generalization capability.

The last parameter that we dealt with was the goal. The goal is used to stop the training in a definite error value. As mentioned above the other parameter we used to stop the training was epoch number. If it reaches the epoch number before it reaches the goal, the training stops. If it reaches the goal before it reaches the epoch number, the training also stops. Because of this, goal and the epoch number must be decided carefully.

In addition, some operations on the data were done. One of these operations was giving the data in a random way. This was because the type of the data. Our data was in a sequential order. Because of this, when we gave the data to the system directly, it was overtraining and lost the ability of generalization. Then, when we gave a new data to the system, the system was not able to learn it in a good way. As a solution, we gave the data in a random order. By this way, the system not only learnt the data we gave, but also it recognized the new data given. Then, we can say that training is more successful when the data is given in a random way.

As happened in the RBF neural network, the outputs that are defined as red, blue and yellow of the net have definite values. After training the net, the outputs of the data used in the test set were going to the output values, but not exactly the same. Because of this, the error rate was very high. In order to overcome this problem, the resultant outputs were rounded to the closest output value. By this way, the error rate of the system was decreased in an acceptable value.

After these operations the parameter we changed was the input number. By increasing the input number from 250 to 400 we observed the positive changes in the

system. The figures from 5.43 to 5.48 and explanations of these figures are given below for this application.

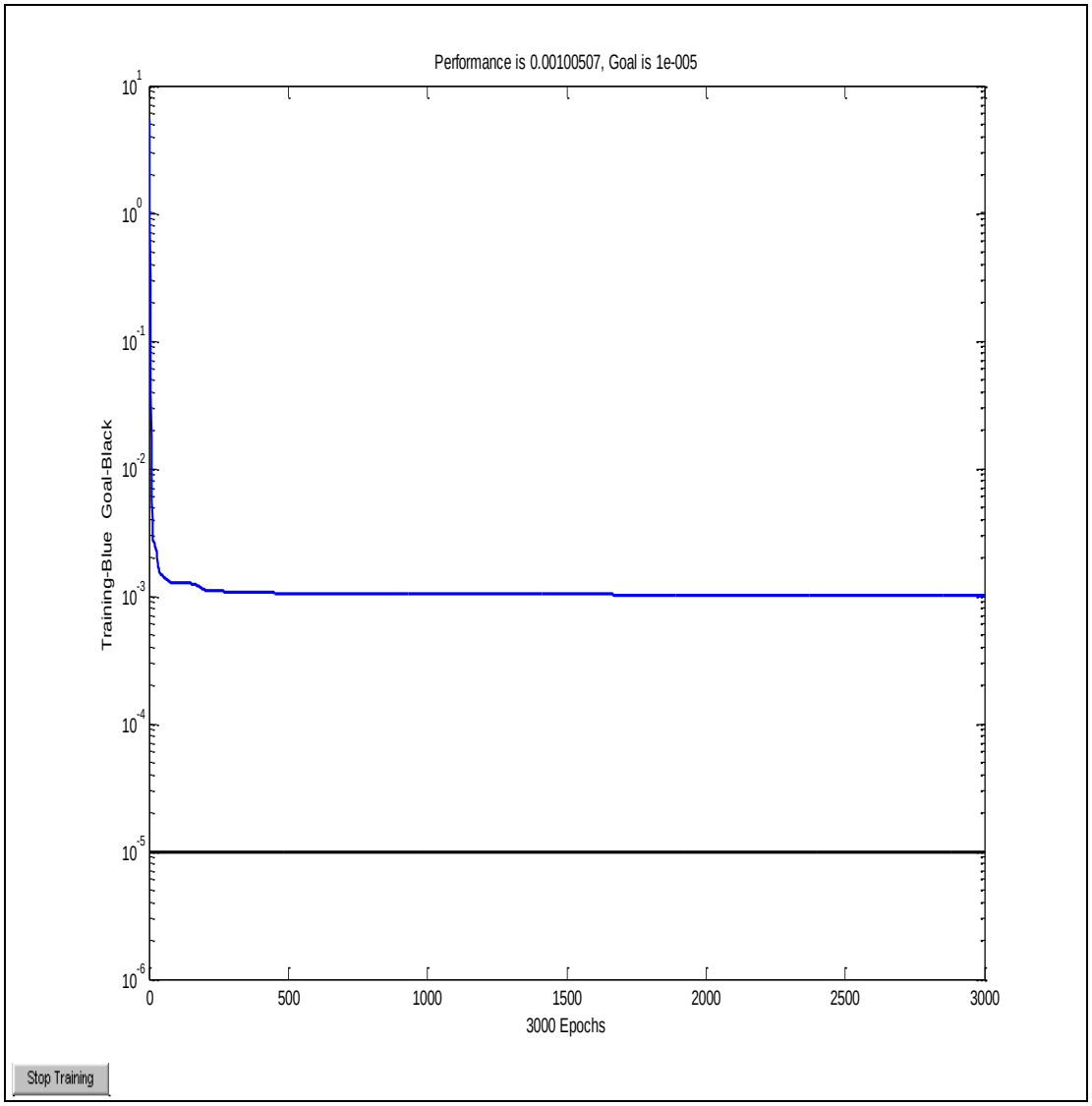

Figure 5.43 The training performance of 15 hidden neurons and 250 inputs

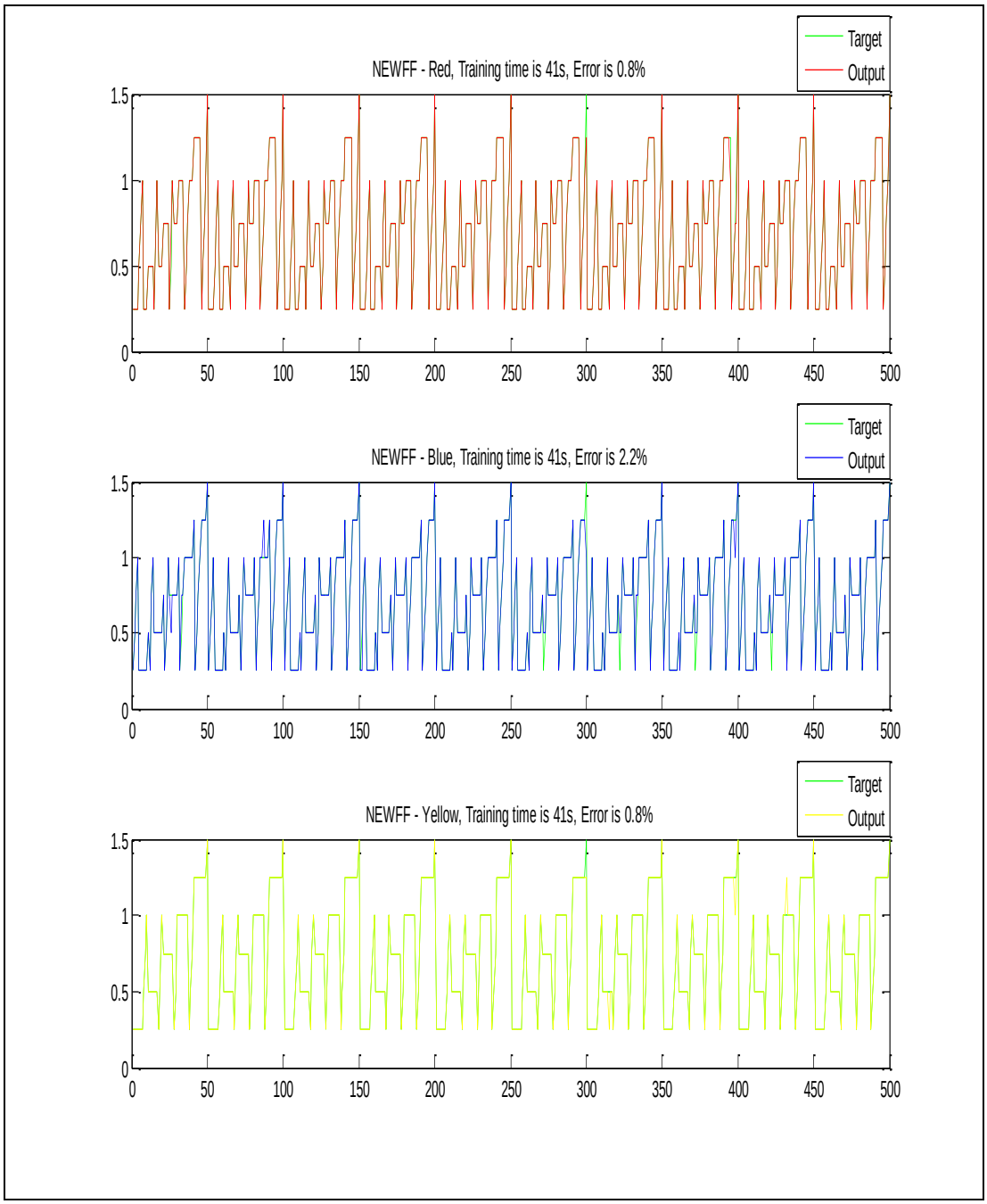

Figure 5.44 The output distribution after training with 15 hidden neurons and 250 inputs

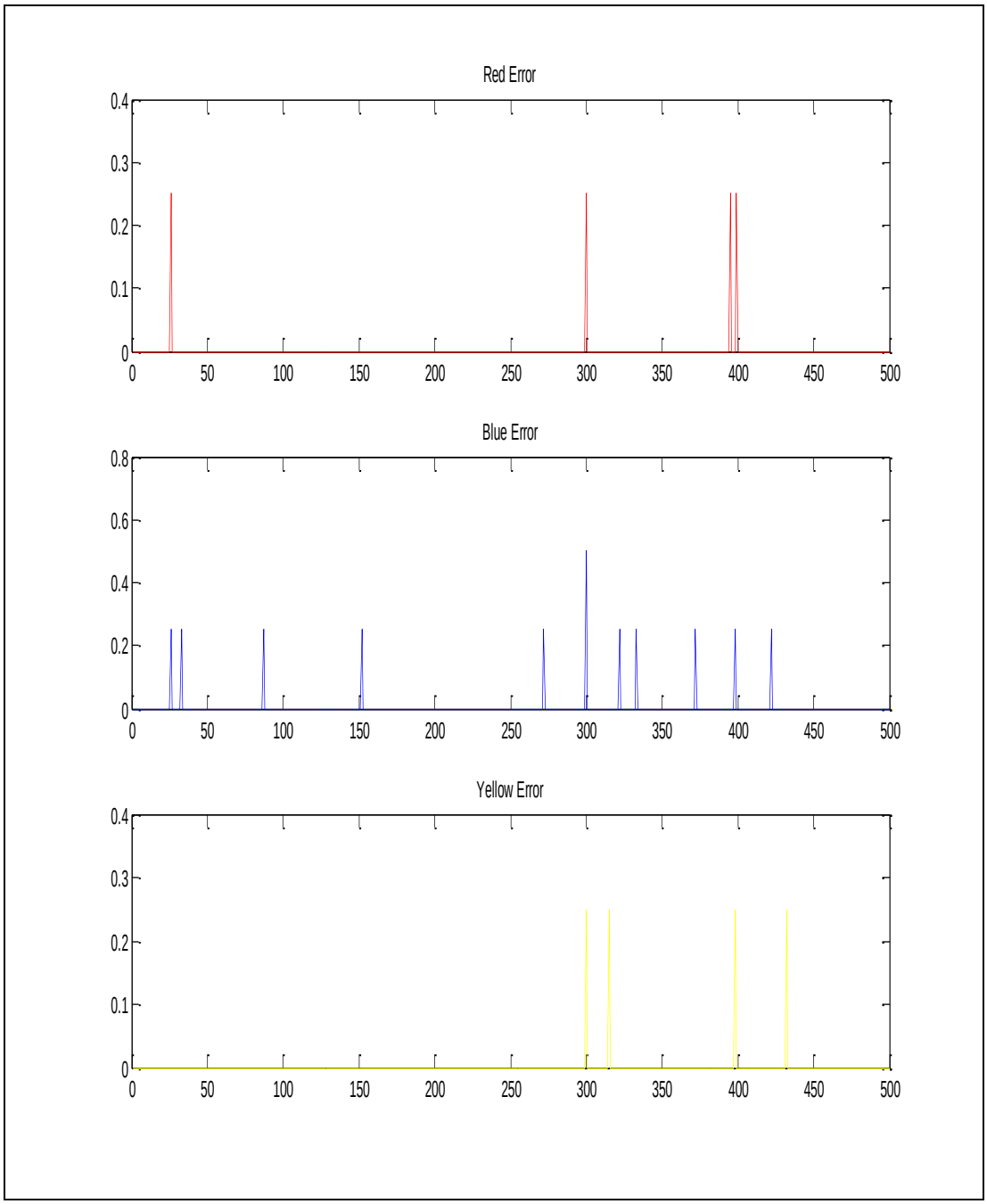

Figure 5.45 The error values of training with 15 hidden neurons and 250 inputs

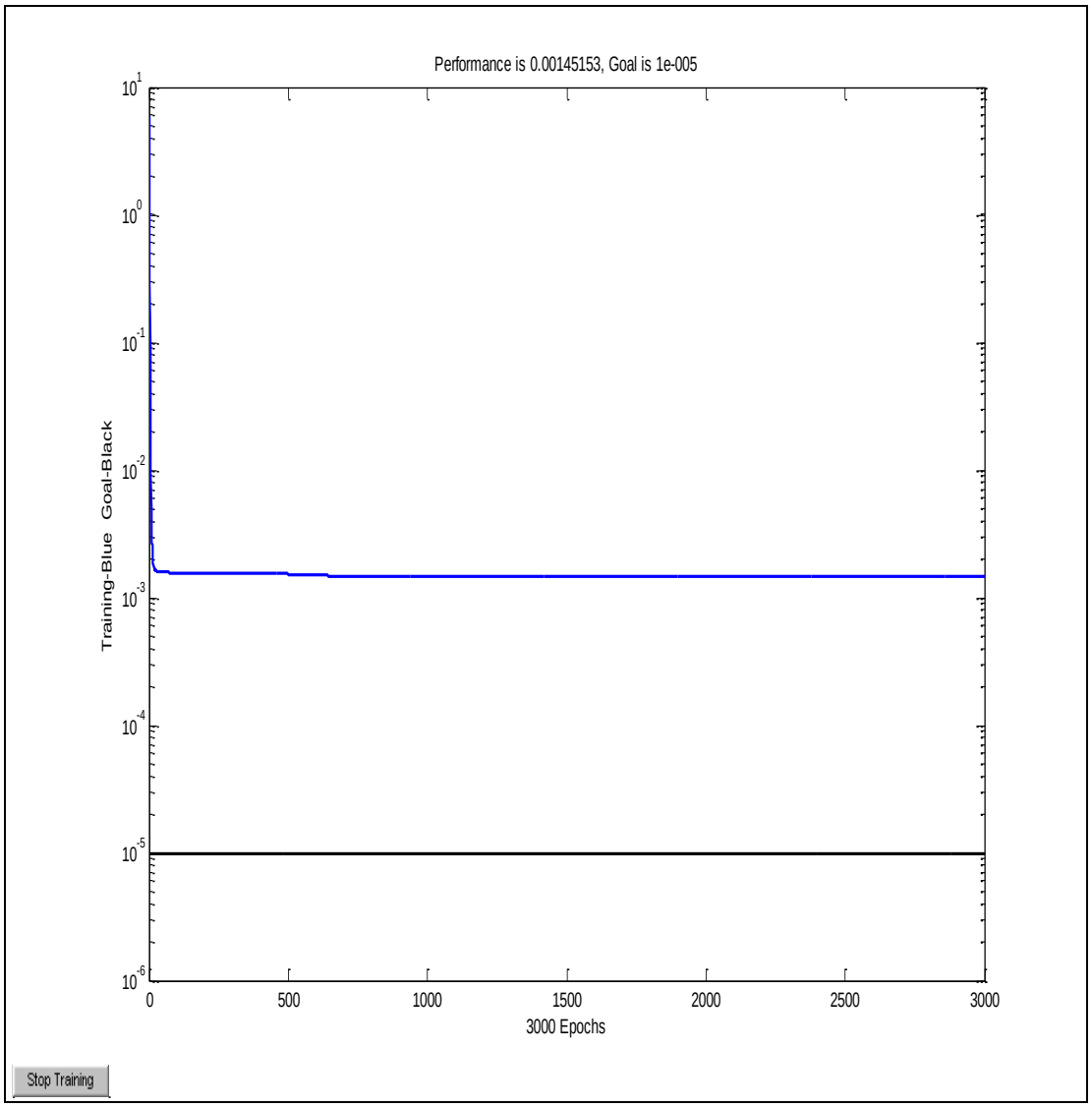

Figure 5.46 The training performance of 15 hidden neurons and 400 inputs

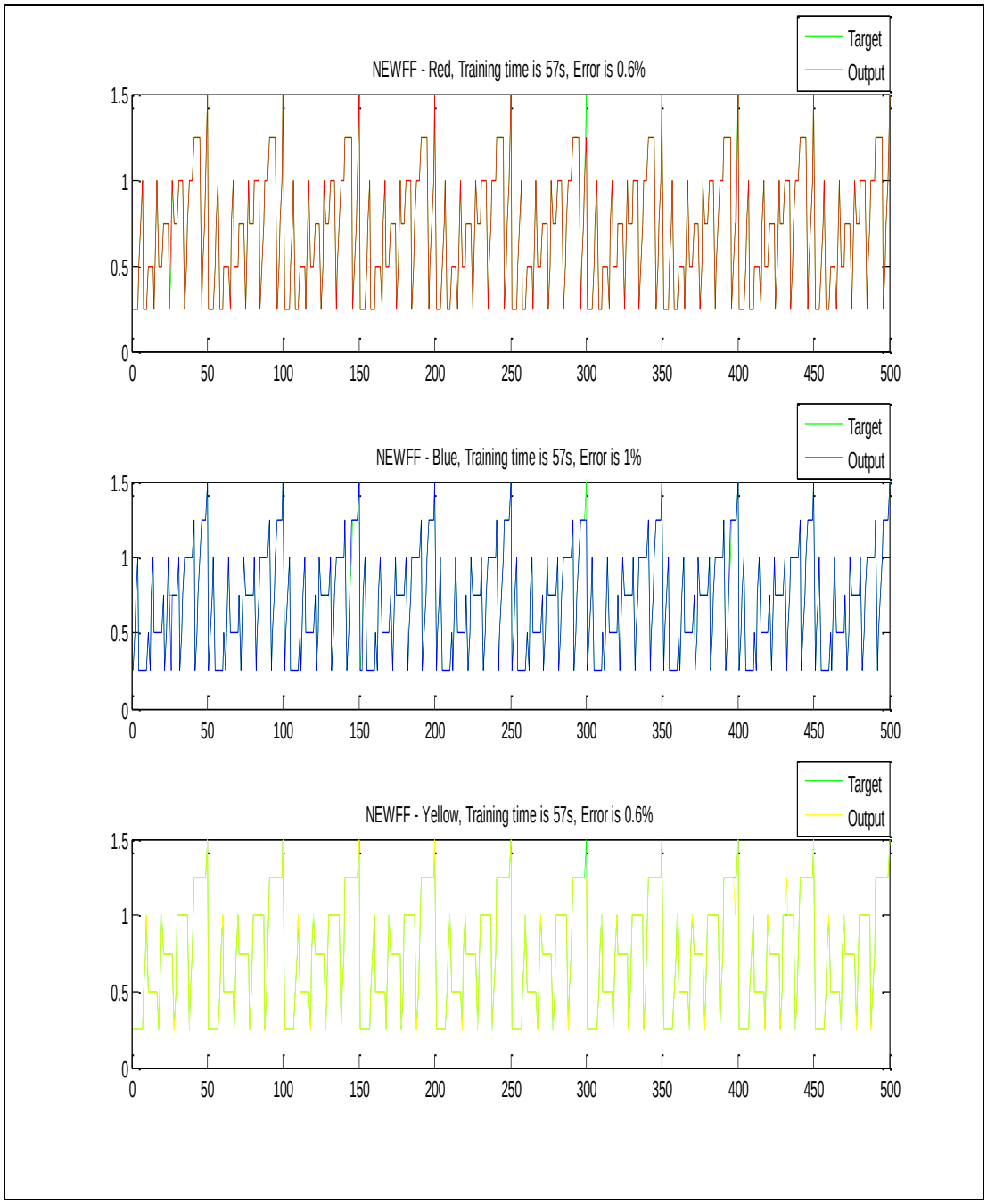

Figure 5.47 The output distribution after training with 15 hidden neurons and 400 inputs

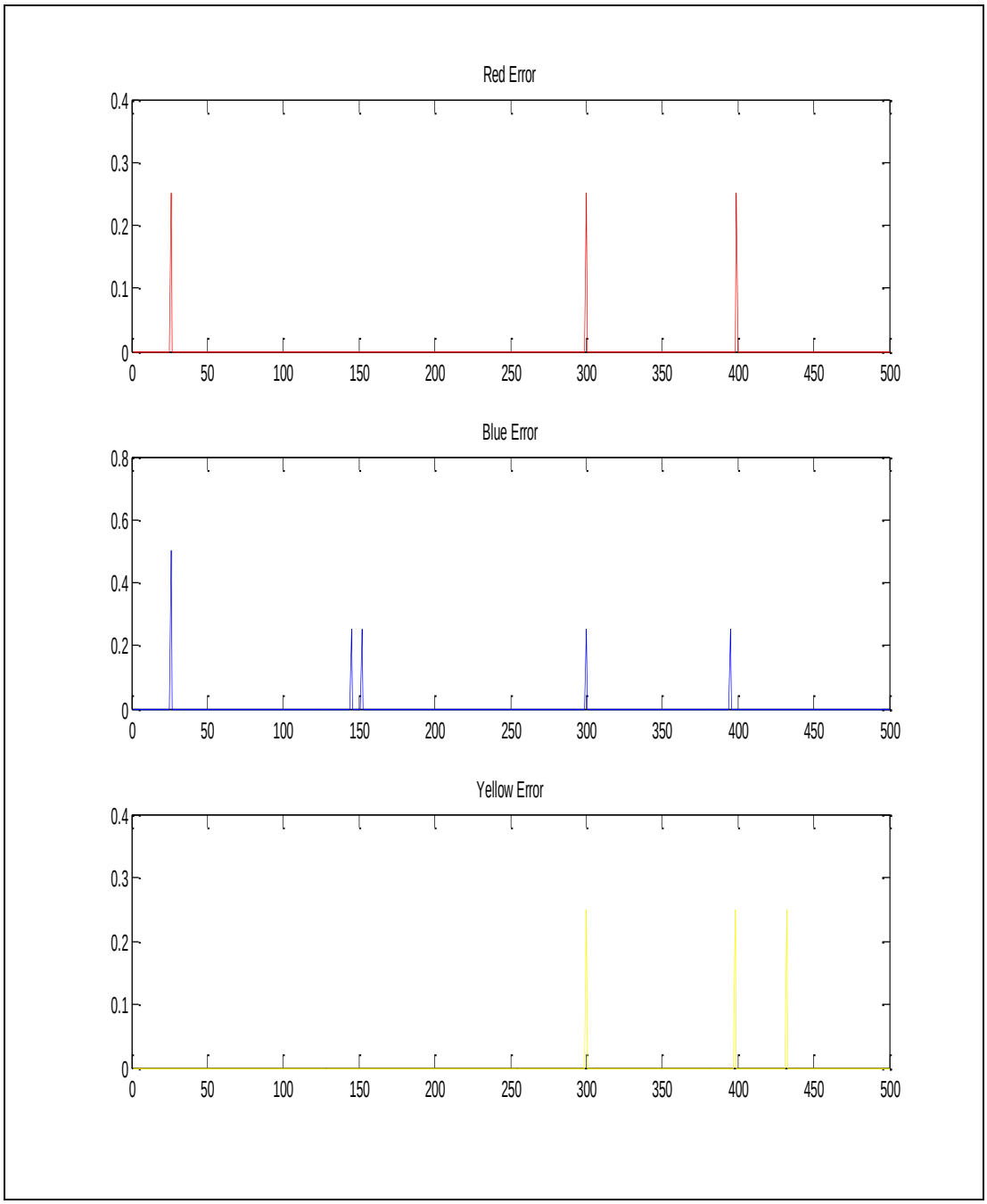

Figure 5.48 The error values of training with 15 hidden neurons and 400 inputs

In the training input number was increased from 250 to 400. It was seen that the time of training was increased from 41 sec to 57 sec. However, the training performance was increased and error percentages were decreased.

This experiment shows that, increasing the training input number affects the systems well, although it increases the training time.

After these operations at last, it was decided to add a second hidden layer to the system. The figures from 5.49 to 5.57 and explanations of these figures are given below for this application.

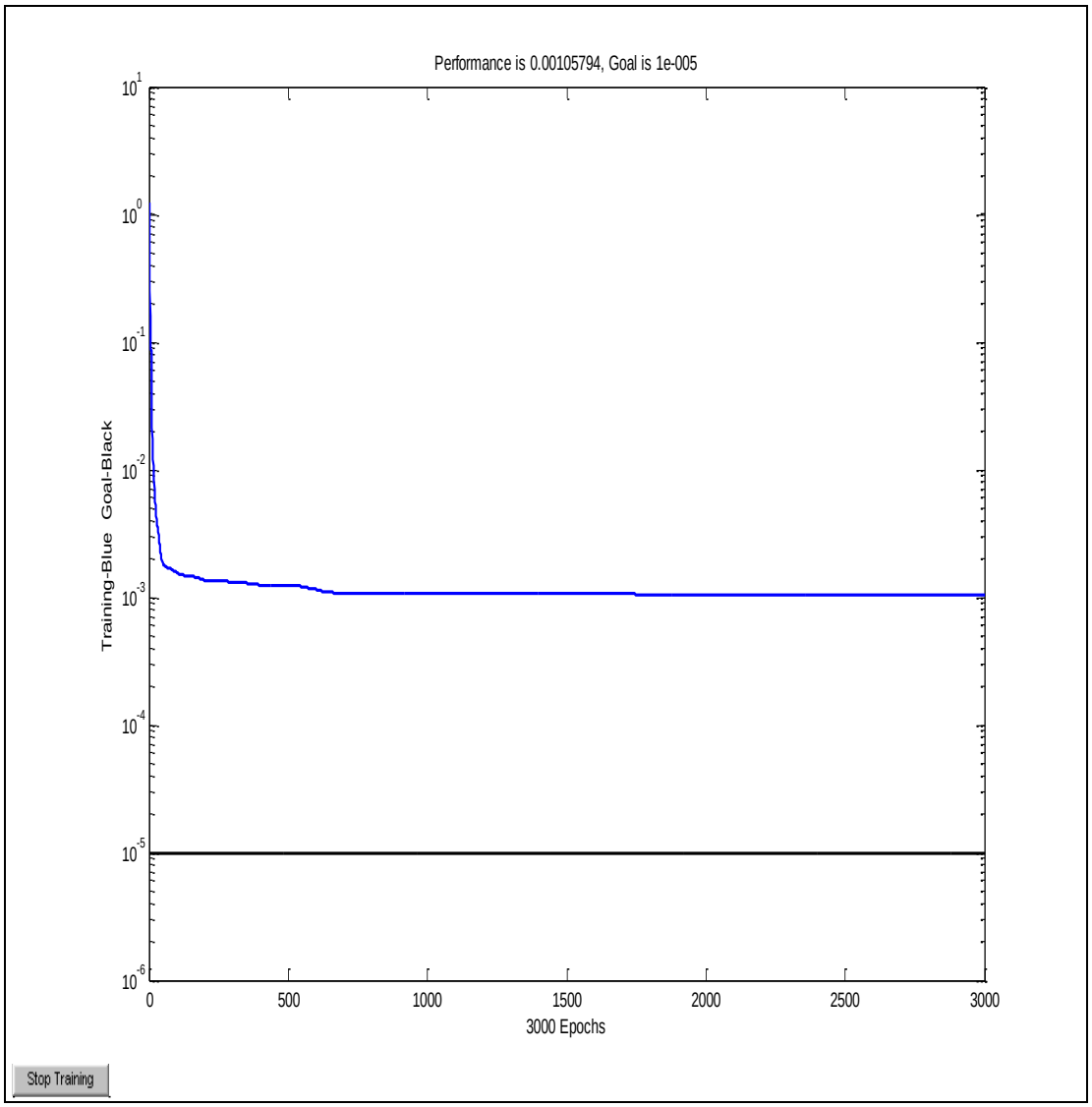

Figure 5.49 The training performance of the system with 2 hidden layers with neurons numbers of 15 at the first layer and 5 at the second layer

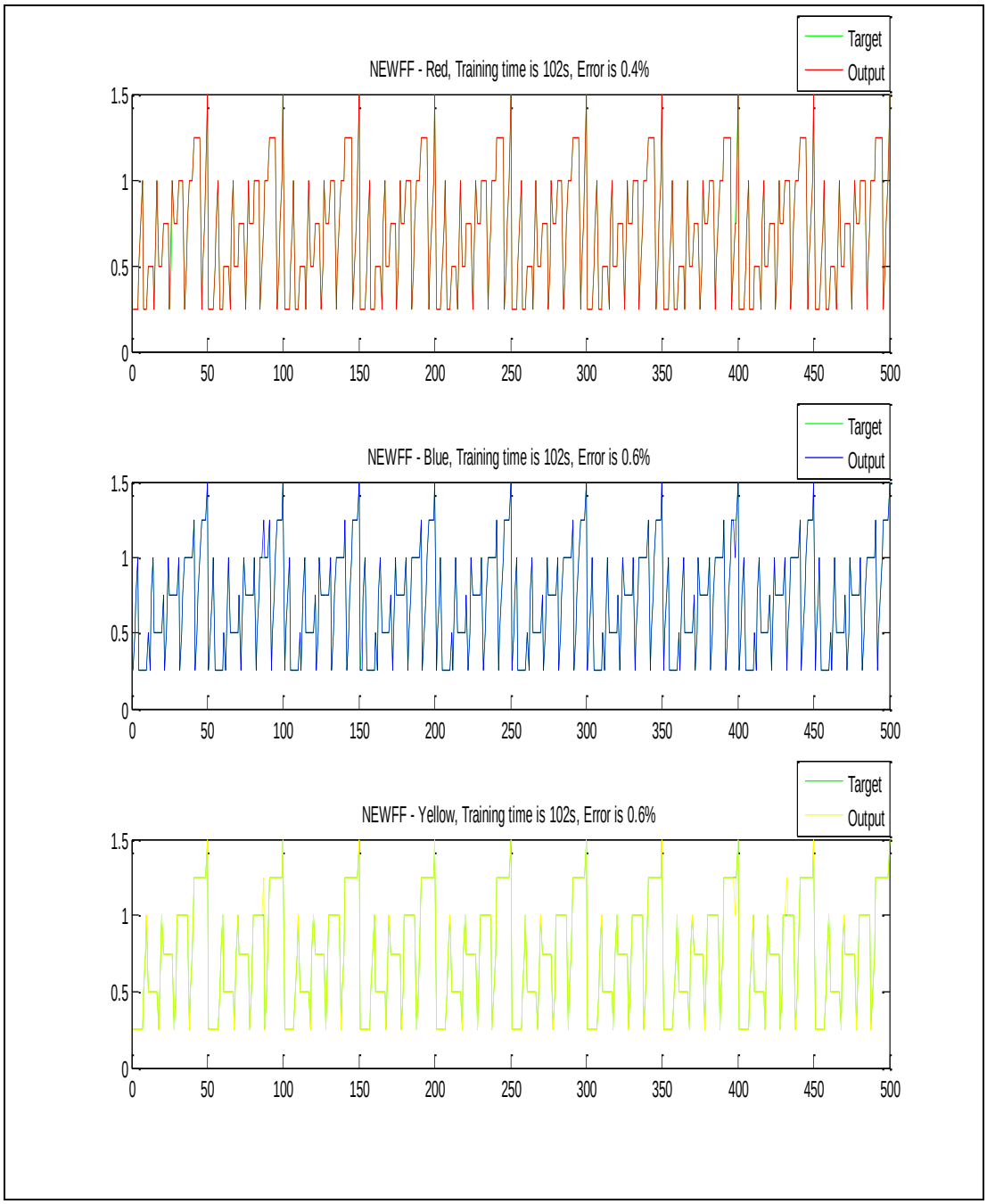

Figure 5.50 The output distribution after training with 2 hidden layers with neurons numbers of 15 at the first layer and 5 at the second layer

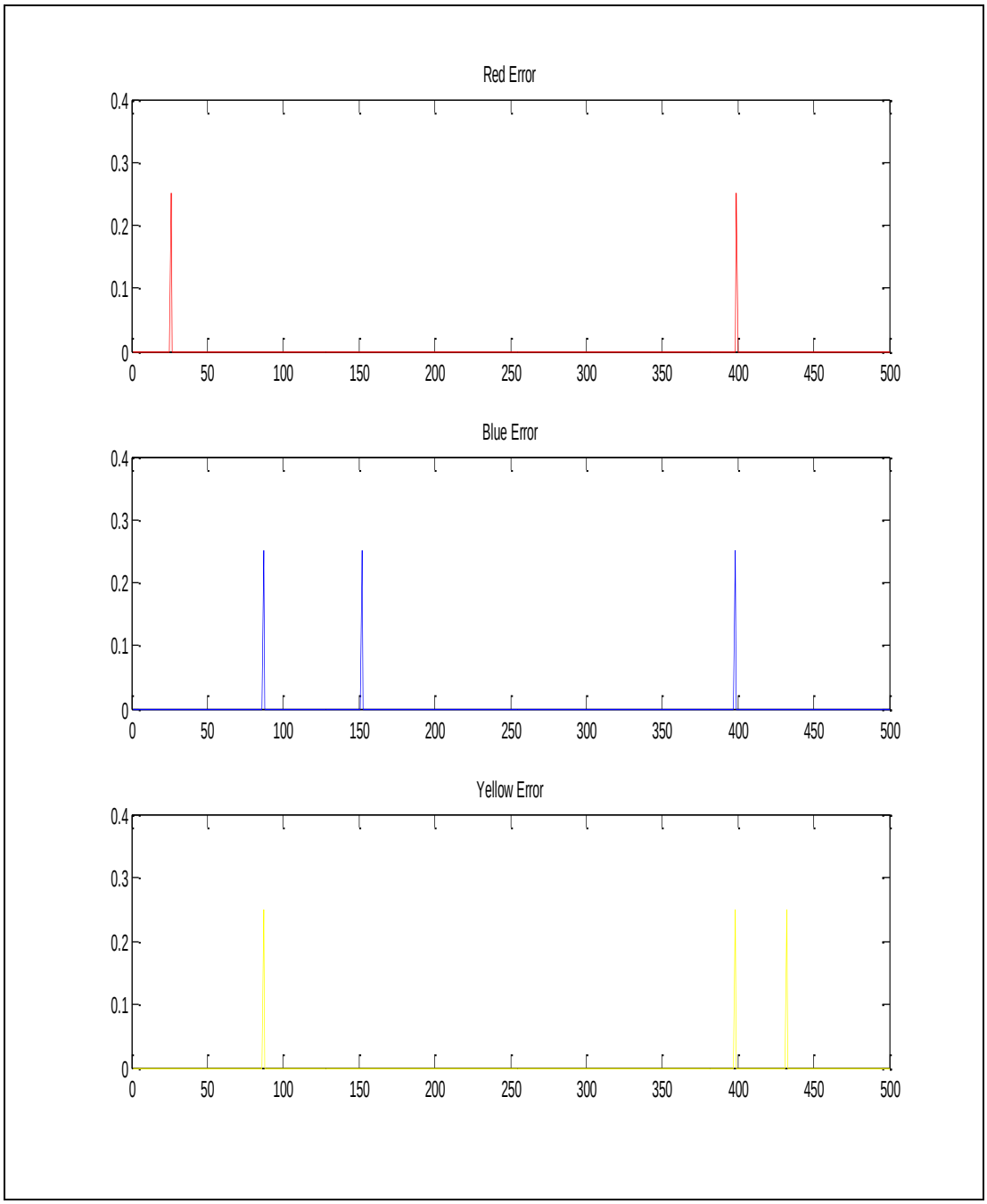

Figure 5.51 The error values of training with 2 hidden layers with neurons numbers of 15 at the first layer and 5 at the second layer

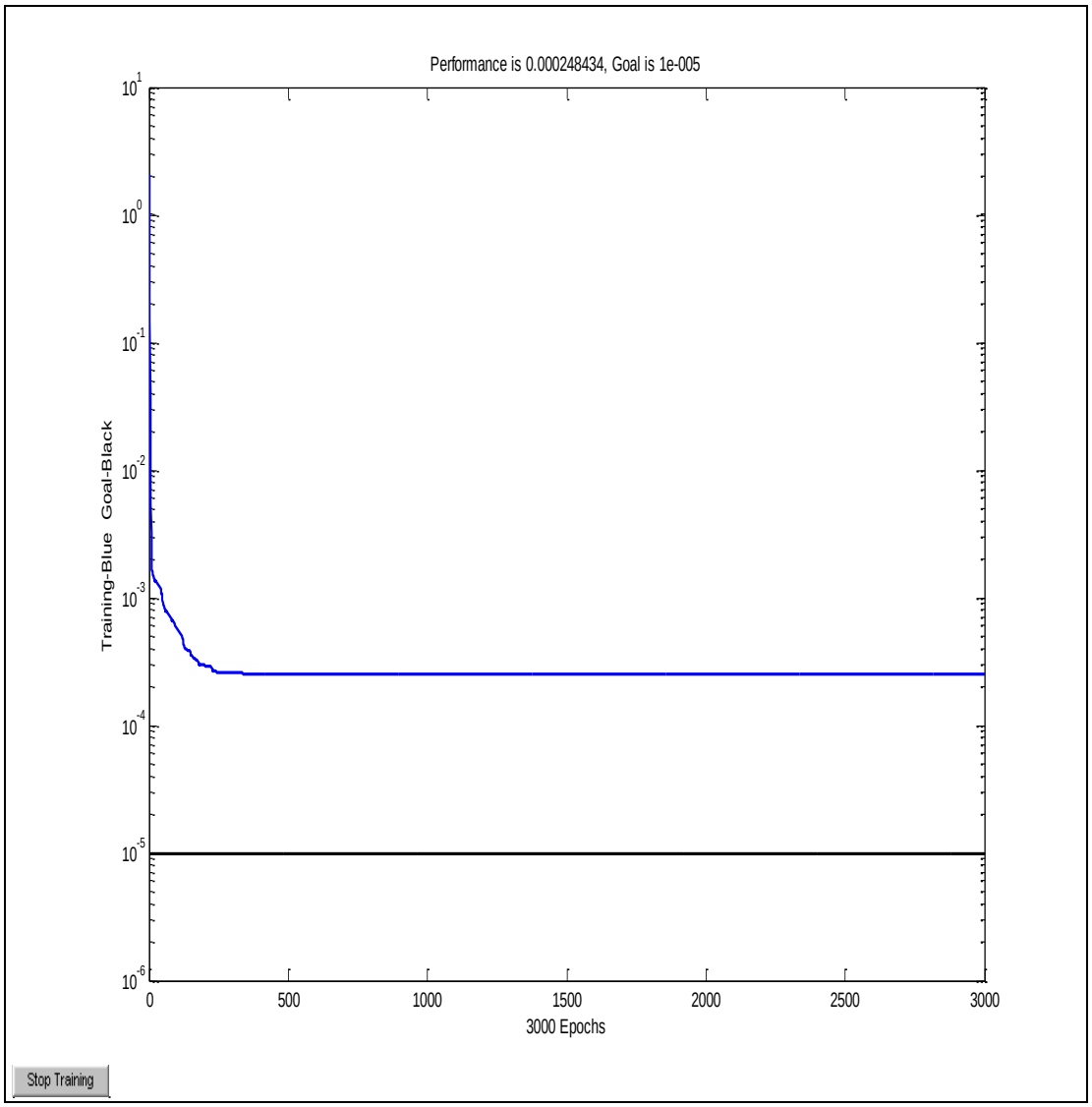

Figure 5.52 The training performance of the system with 2 hidden layers with neurons numbers of 15 at the first layer and 10 at the second layer

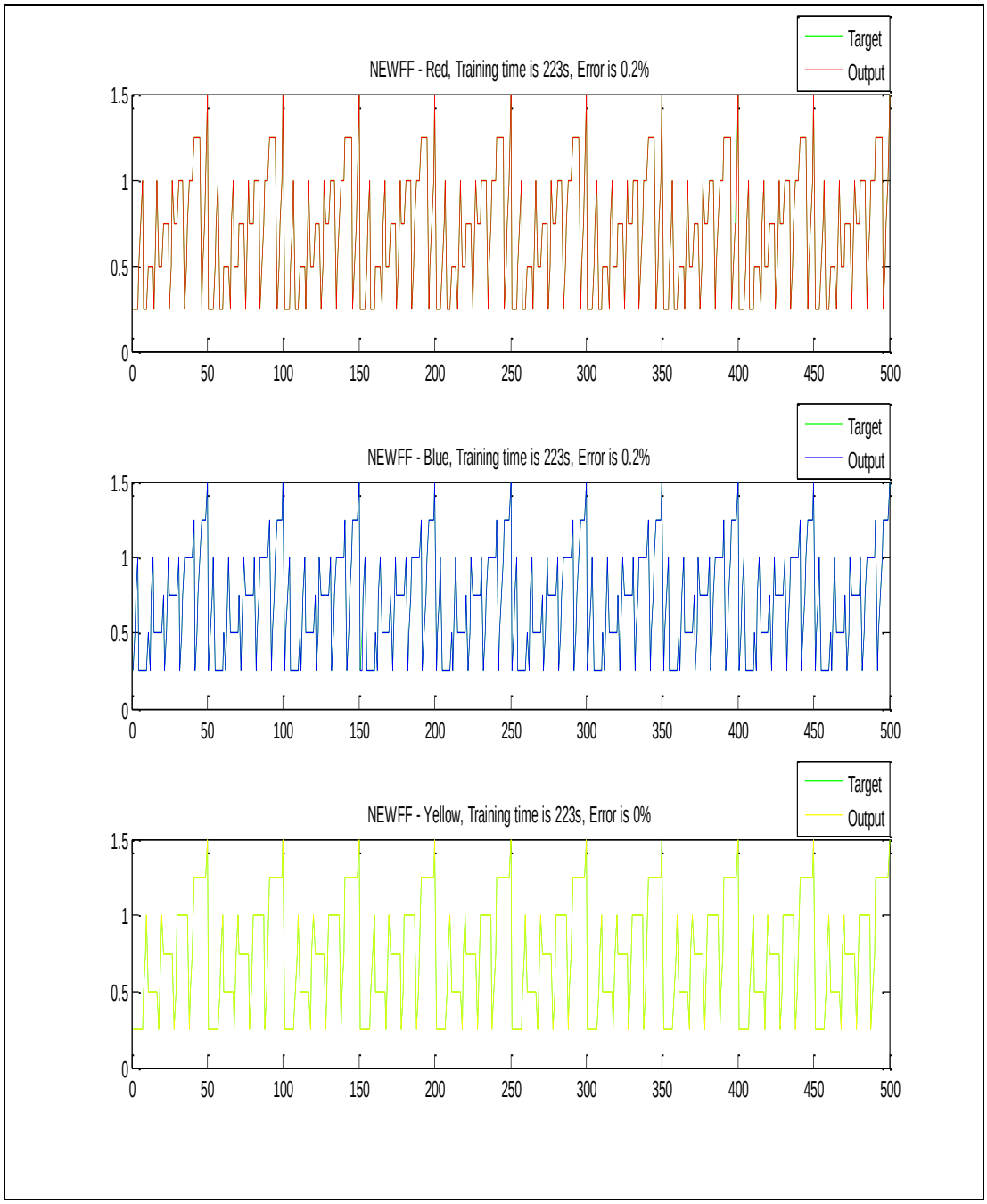

Figure 5.53 The output distribution after training with 2 hidden layers with neurons numbers of 15 at the first layer and 10 at the second layer

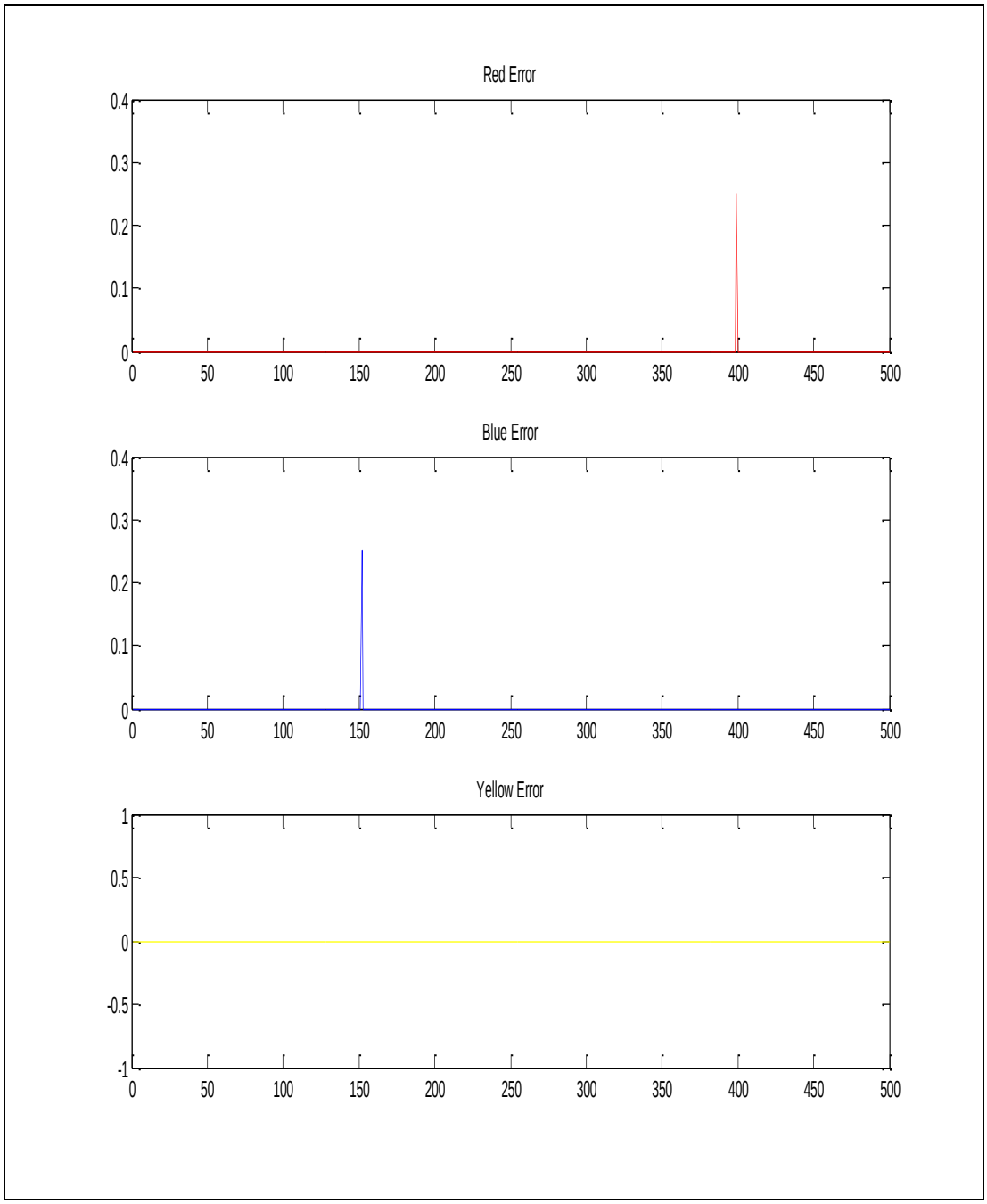

Figure 5.54 The error values of training with 2 hidden layers with neurons numbers of 15 at the first layer and 10 at the second layer

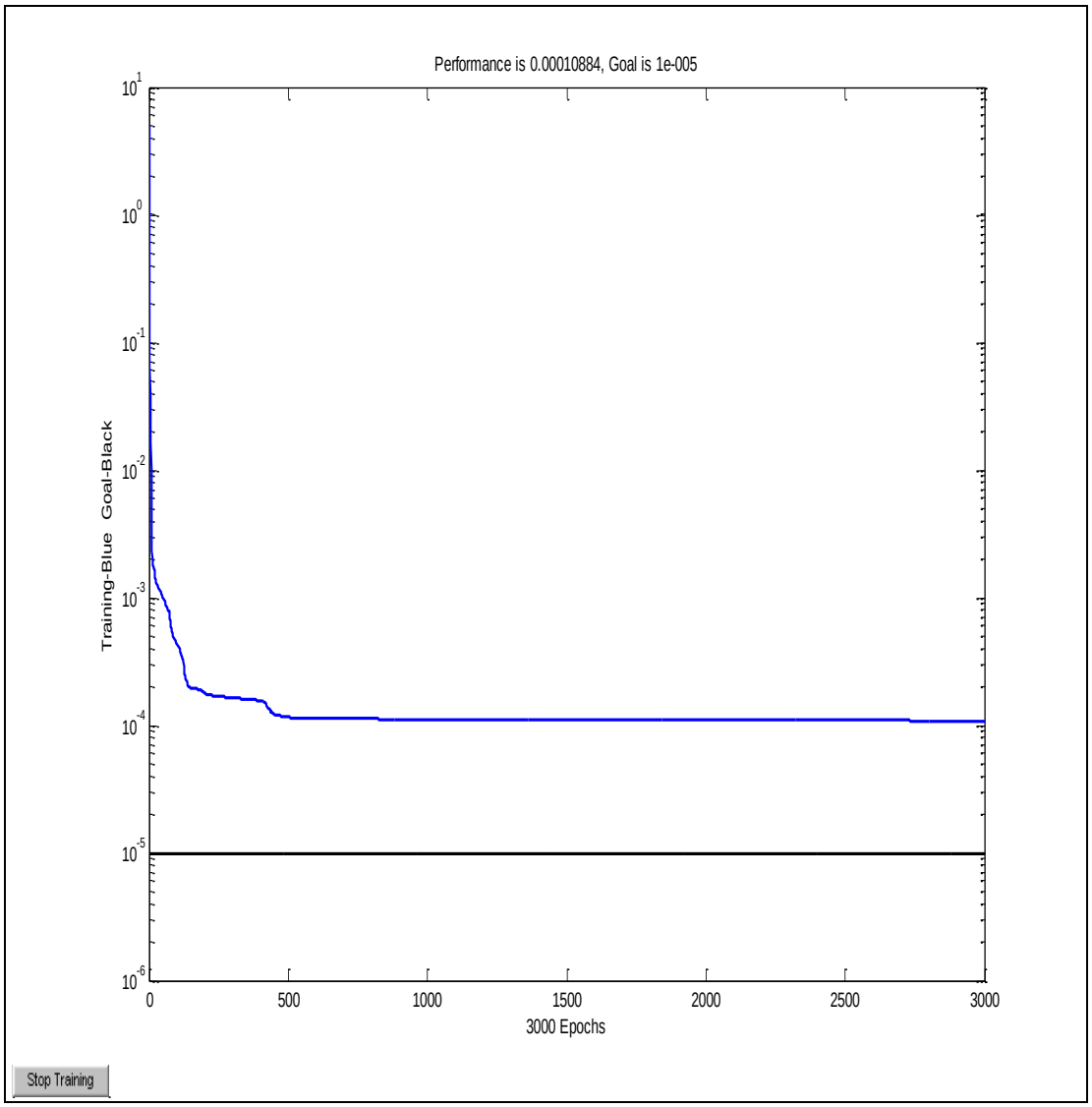

Figure 5.55 The training performance of the system with 2 hidden layers with neurons numbers of 15 at the first layer and 15 at the second layer

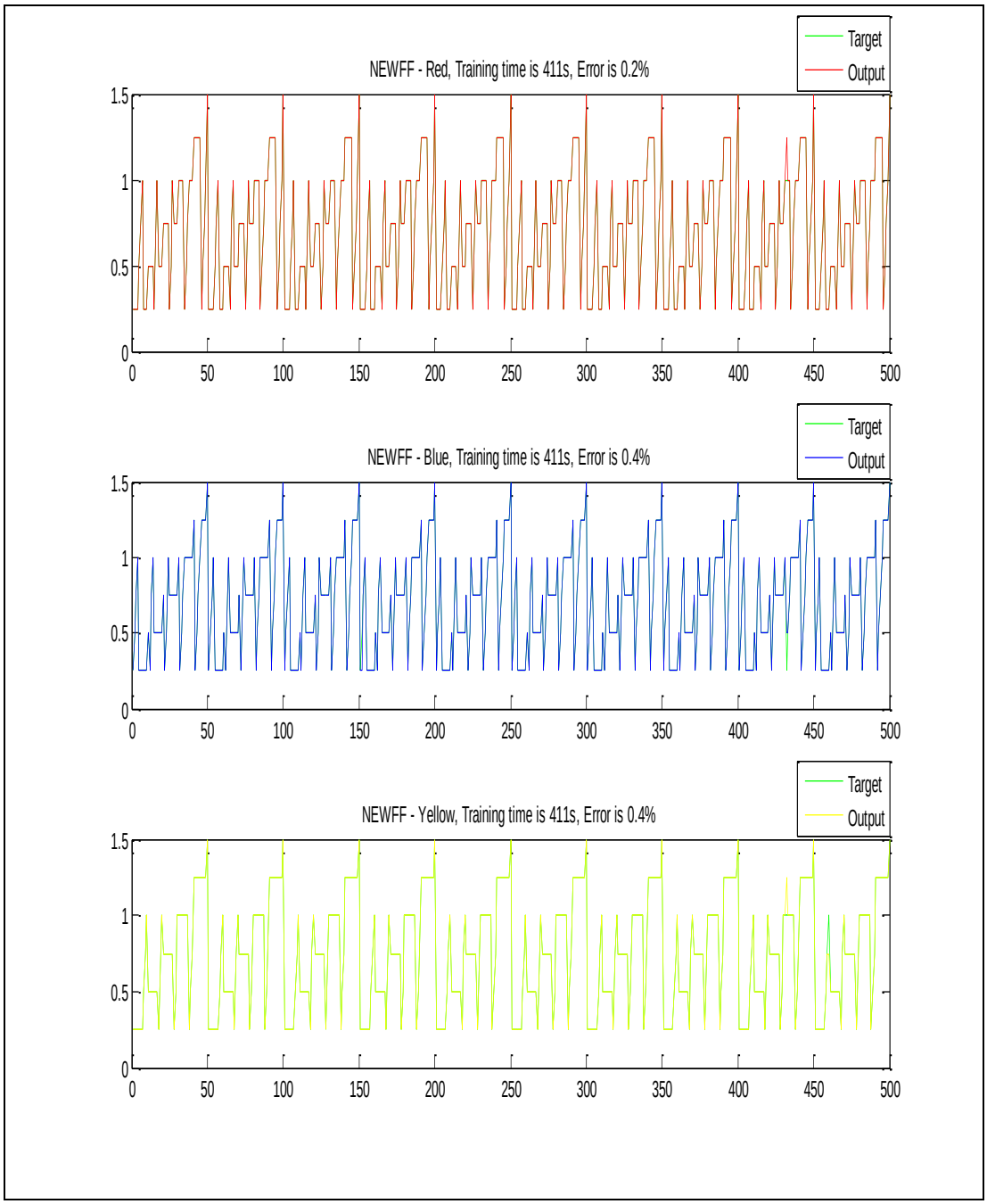

Figure 5.56 The output distribution after training with 2 hidden layers with neurons numbers of 15 at the first layer and 15 at the second layer

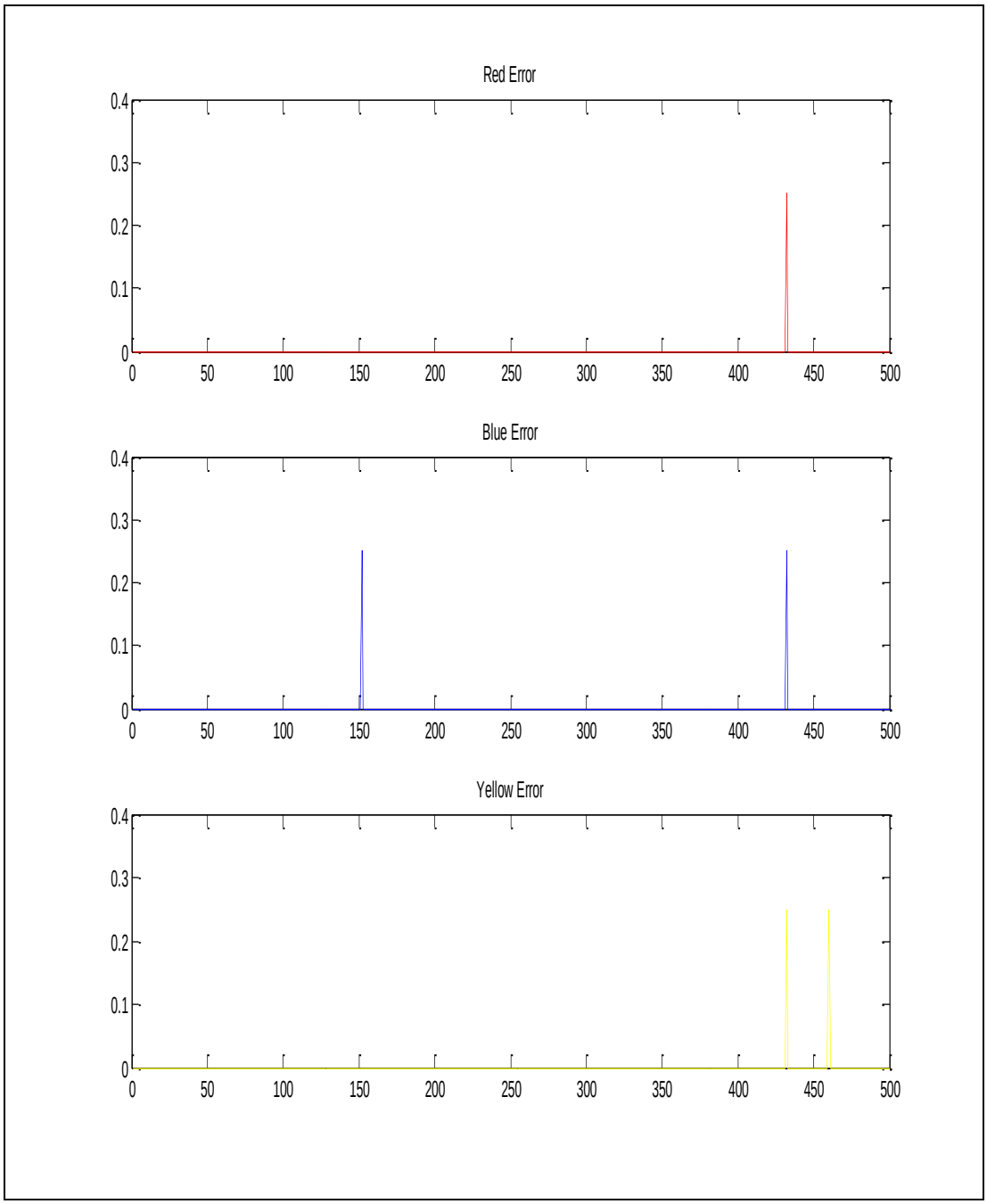

Figure 5.57 The error values of training with 2 hidden layers with neurons numbers of 15 at the first layer and 15 at the second layer

In the training first of all, the second hidden layer neuron number was increased from 5 to 10. It was seen that the time of training was increased from 102 sec to 223 sec. However, the training performance was increased and error percentages were decreased.

Secondly, it was tried to observe that what will happen if the second hidden layer neuron number was increased from 10 to 15. It was seen that both the training time and error percentages increased. However the training performance of the system was decreased.

These experiments show that increasing the hidden layer neuron number too much does not mean that the system will have a good performance of training. It has a limitation.

At last, with two hidden layers and different hidden neuron numbers training was done and the figures 5.58, 5.59 and 5.60, also explanations of these figures are given below for this application.

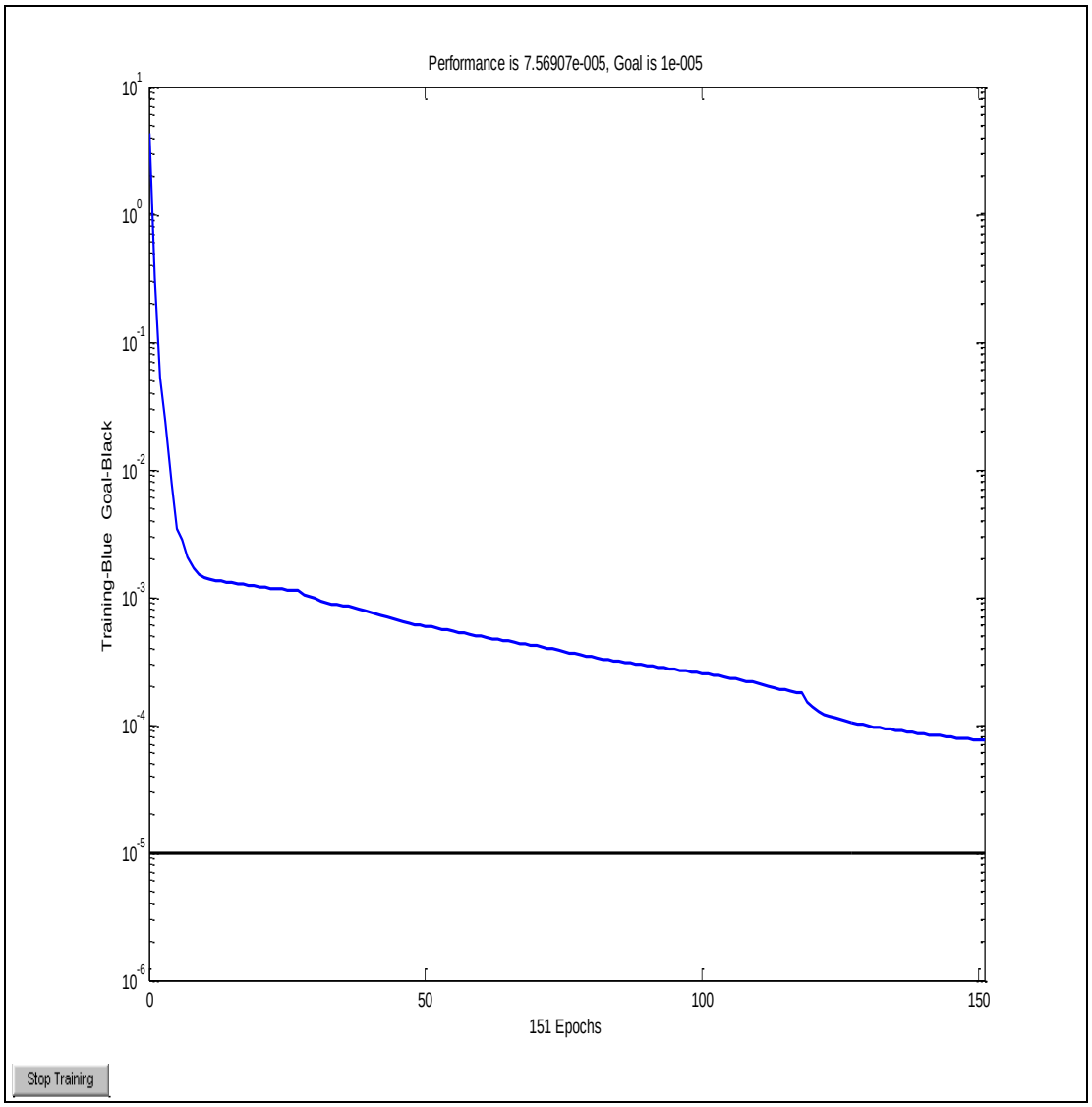

Figure 5.58 The training performance of the system with 2 hidden layers with neurons numbers of 20 at the first layer and 25 at the second layer

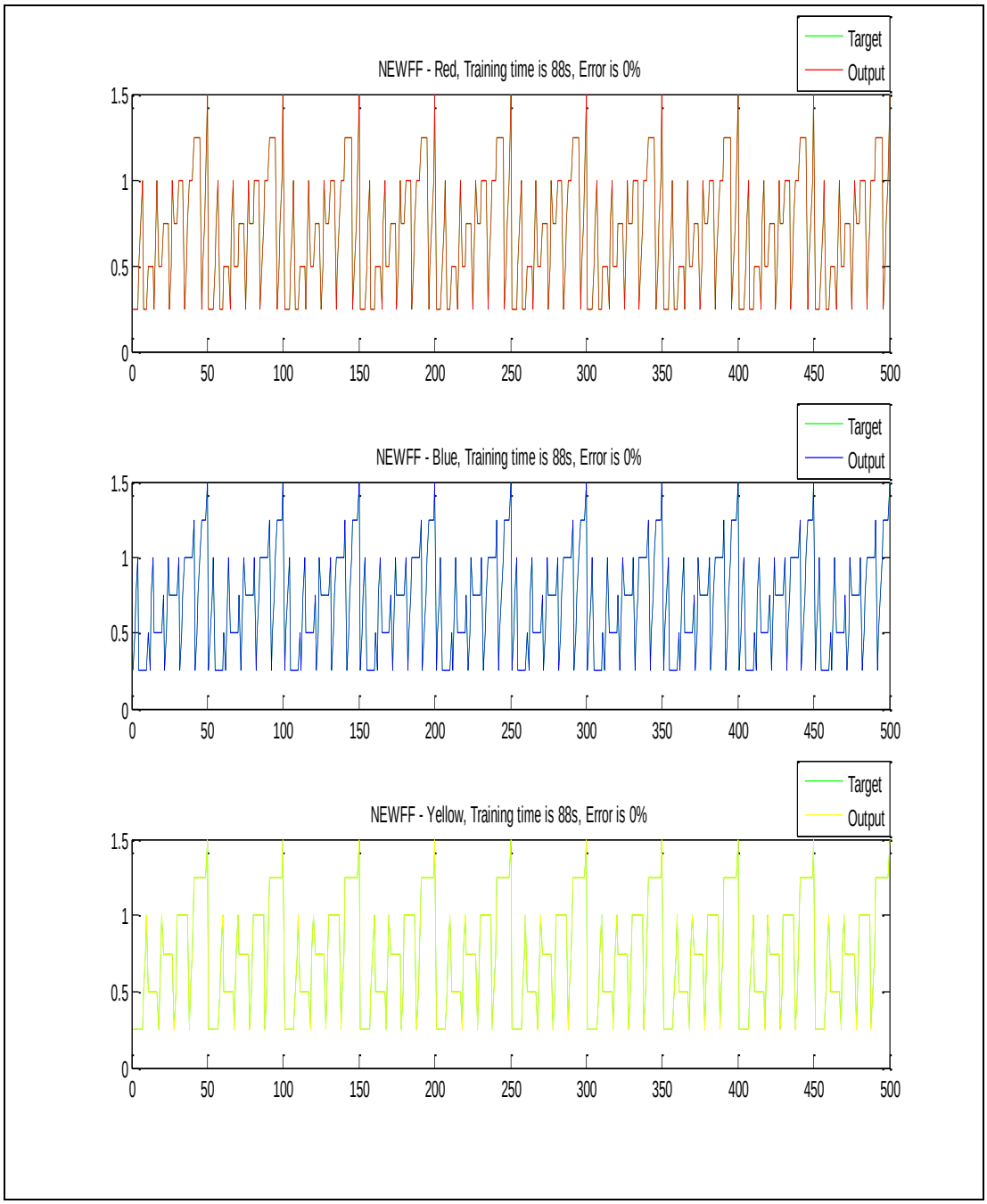

Figure 5.59 The output distribution after training with 2 hidden layers with neurons numbers of 20 at the first layer and 25 at the second layer

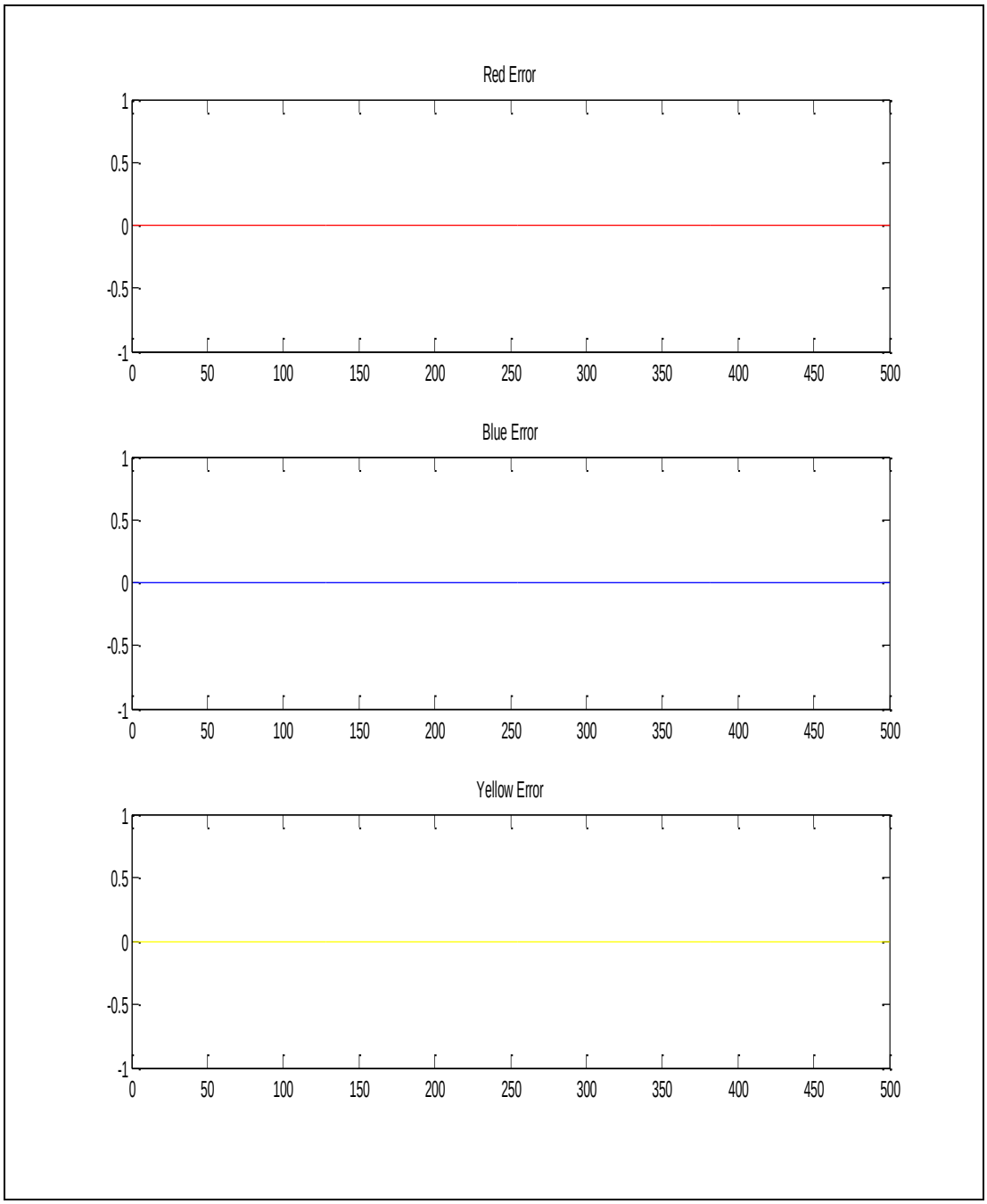

Figure 5.60 The error values of training with 2 hidden layers with neurons numbers of 20 at the first layer and 25 at the second layer

It was seen in the previous applications that increasing the number of hidden layers to 2 increased the performance of the system. The epoch number of 3000 was used in this application generally. The neuron numbers in the hidden layers were chosen as 20 and 25 after a lot of experiments for these data. With the help of these experiments at last 2 hidden layers were chosen with neuron numbers of 20 and 25. The epoch number was chosen as 3000. The system was stopped before overtraining

in the epoch number of 151 as it can be seen from the Figure 5.58. As it is seen from Figure 5.60, the error percentages of all of the three colors were decreased to 0%. This means a success of 100%.

5.3 Applications of Fuzzy Logic

As mentioned in the theoretical part; there are some methods used in the fuzzy logic. In this thesis, anfis was chosen in order to train the network. The reason of this choice was its relationship with artificial neural network. As it can be understood from its name, anfis is a hybrid system which contains artificial neural network and fuzzy logic together. This method also has some advantages and disadvantages as the other methods we used. Its biggest disadvantage is the need for high memory storage. The more the number of membership functions per input increases the system learning. However, this better learning has a cost that it needs a lot of time to converge. Because of this although increase in the membership function affects the system very well, it cannot be possible to increase it too much because of the learning time and memory limitations.

5.3.1 Anfis Applications

In Matlab, the general usage of fuzzy logic is with the command of anfis. In this method various parameters changed and how their results affect the system were observed in the graphs.

There are some membership functions used in the anfis. These are trimf, trapmf, gbellmf, gaussmf, gauss2mf, sigmf, dsigmf, psigmf, pimf, smf, zmf. After long trainings it was decided to use gbellmf, because it is the most general used MF in anfis.

Moreover, it was decided to train the colors red, blue and yellow with different nets. In order to realize this, three different networks were set and for these networks the system was trained.

As it was done in the RBF and MLP, the outputs that are defined as red, blue and yellow of the net have definite values. When we trained the net, the outputs of the data used in the test set were going to the outputs which we did not have. Because of this reason, the error rate was very high. In order to overcome this problem, the resultant outputs were rounded to the closest output value. By this way, the error rate of the system was decreased in an acceptable value.

As mentioned above, there are various parameters that affect the training of anfis. One of them is number of membership functions (MF) per input. When the number of MF per input increases, the training of the system becomes better. However, it has a limitation, because it needs a very huge memory space in the PC system. It is not a realizable thing to increase the number of MF per input to a large extend. Because of this limitation, this parameter is not so changeable for a standard PC. In this thesis, a lot of trials were done and the suitable value decided was 4.

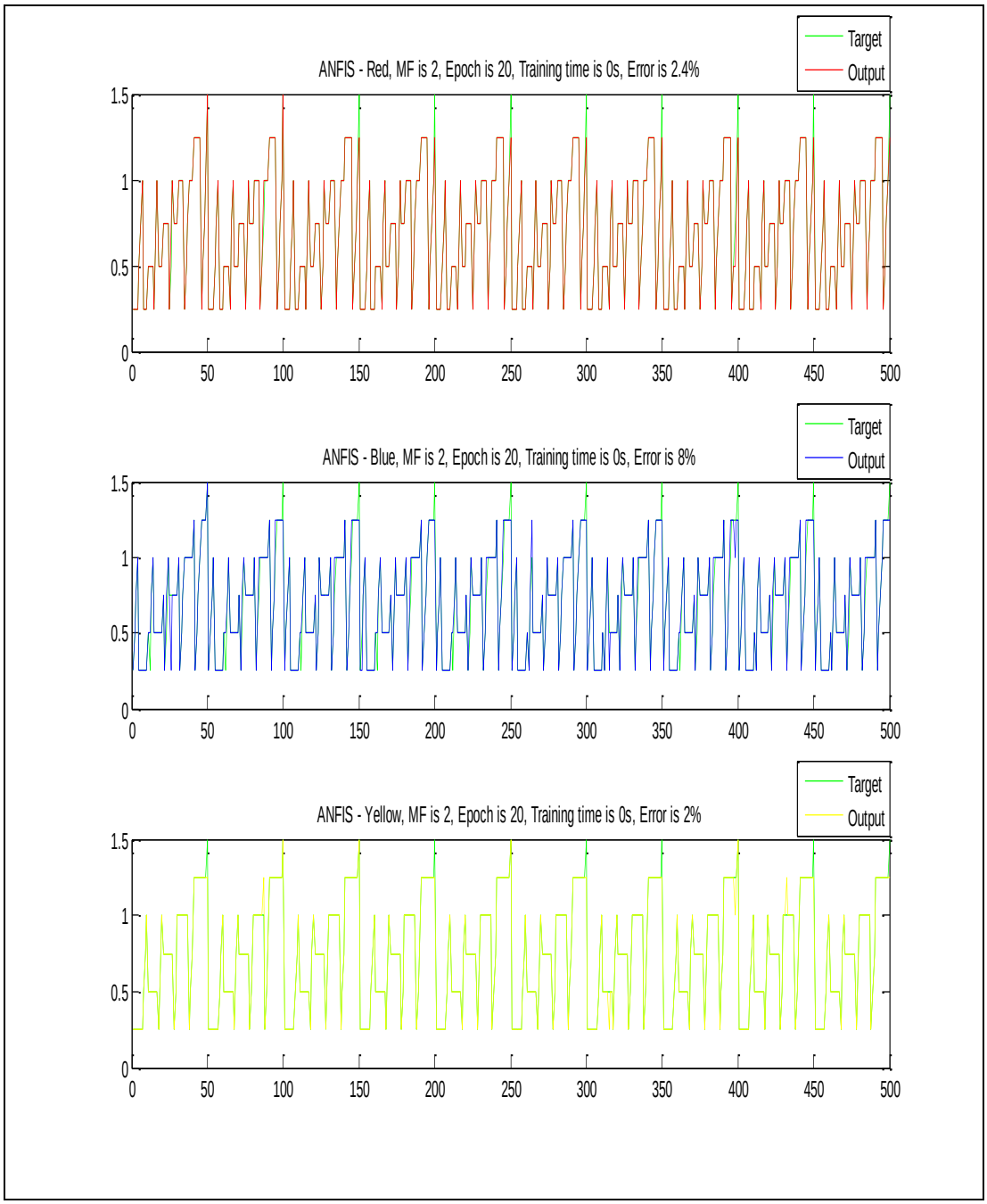

Figure 5.61 The output distribution of MF of 2 and epoch number 20.

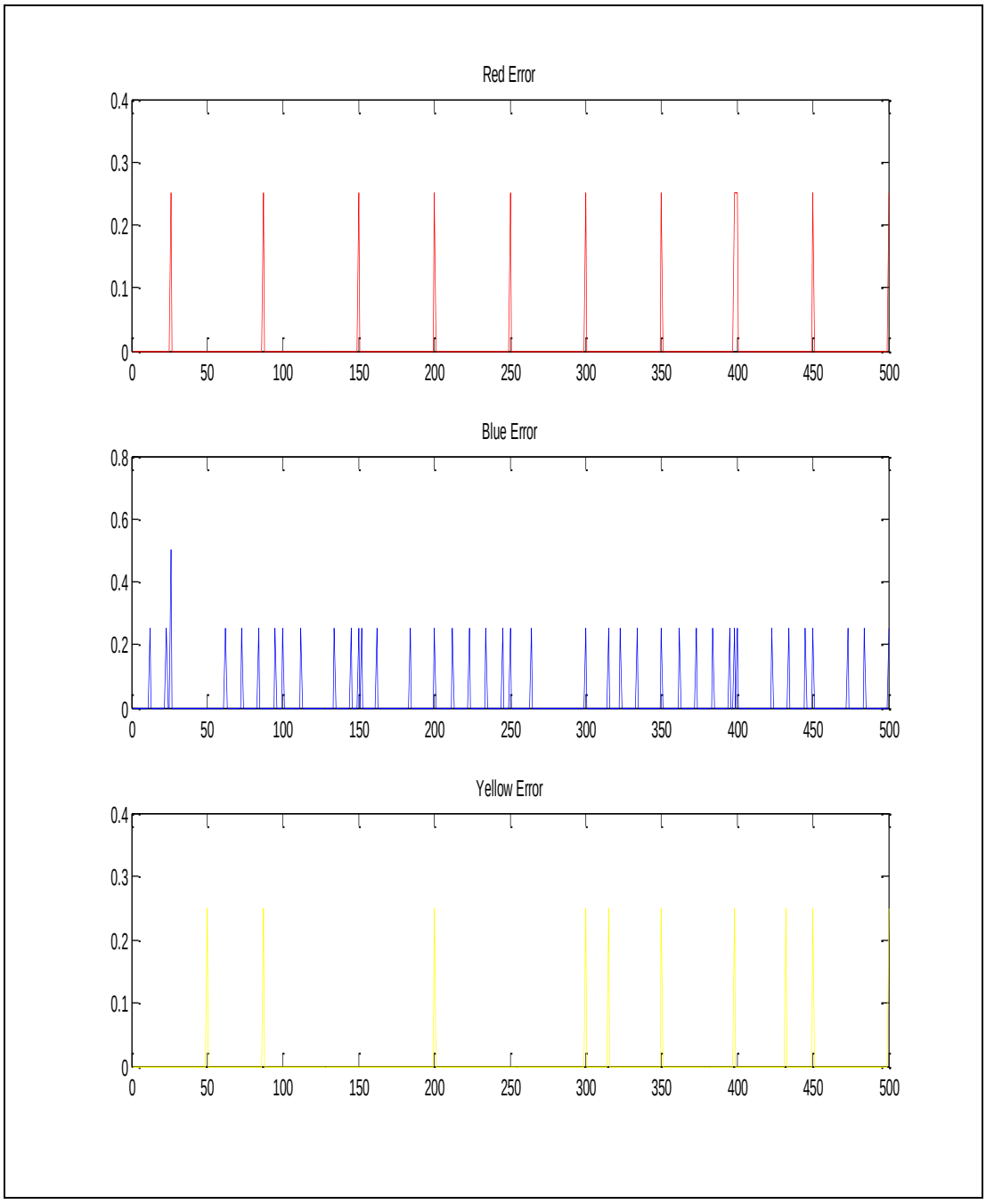

Figure 5.62 The error values for MF of 2

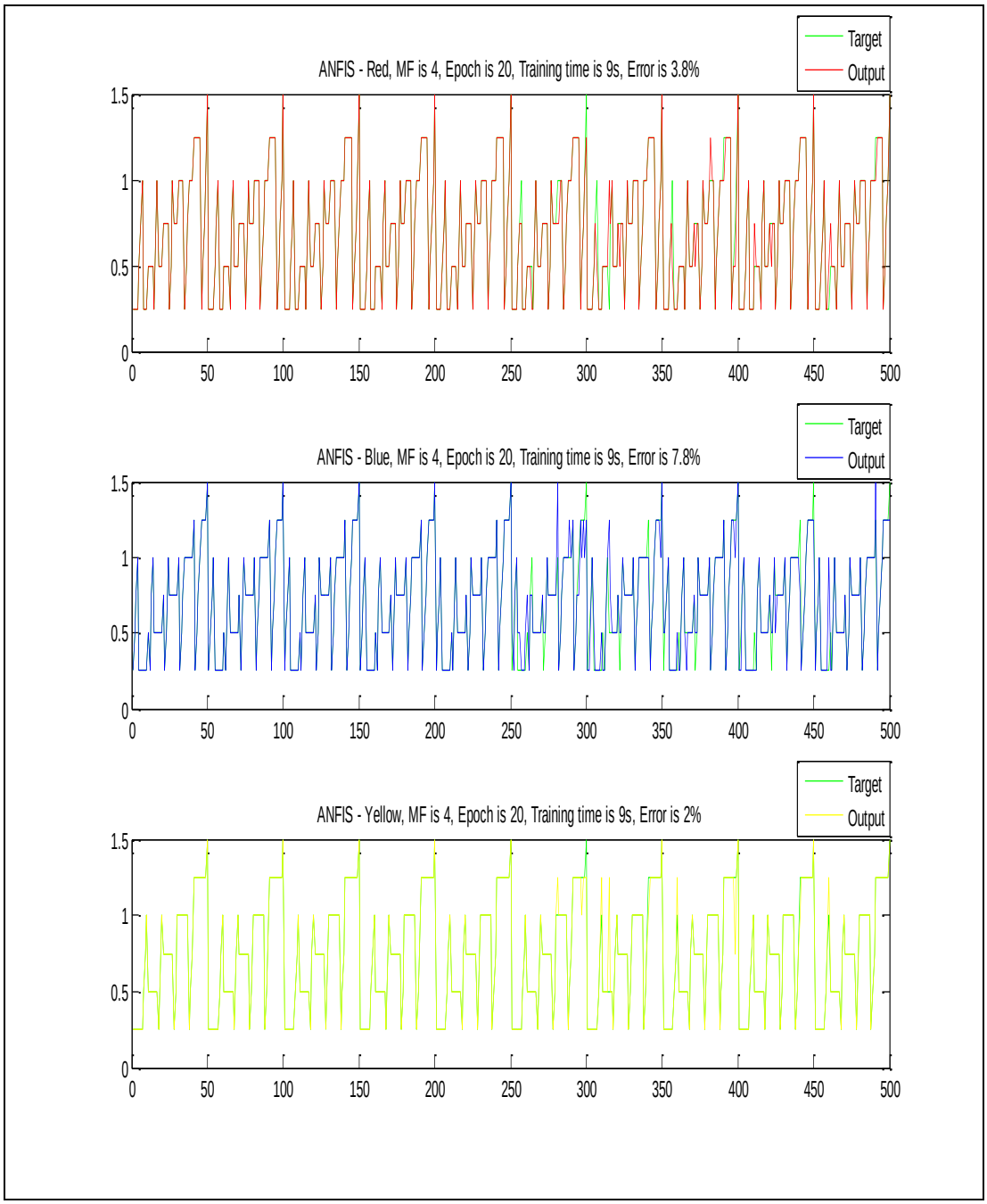

Figure 5.63 The output distribution of MF of 4 and epoch number 20.

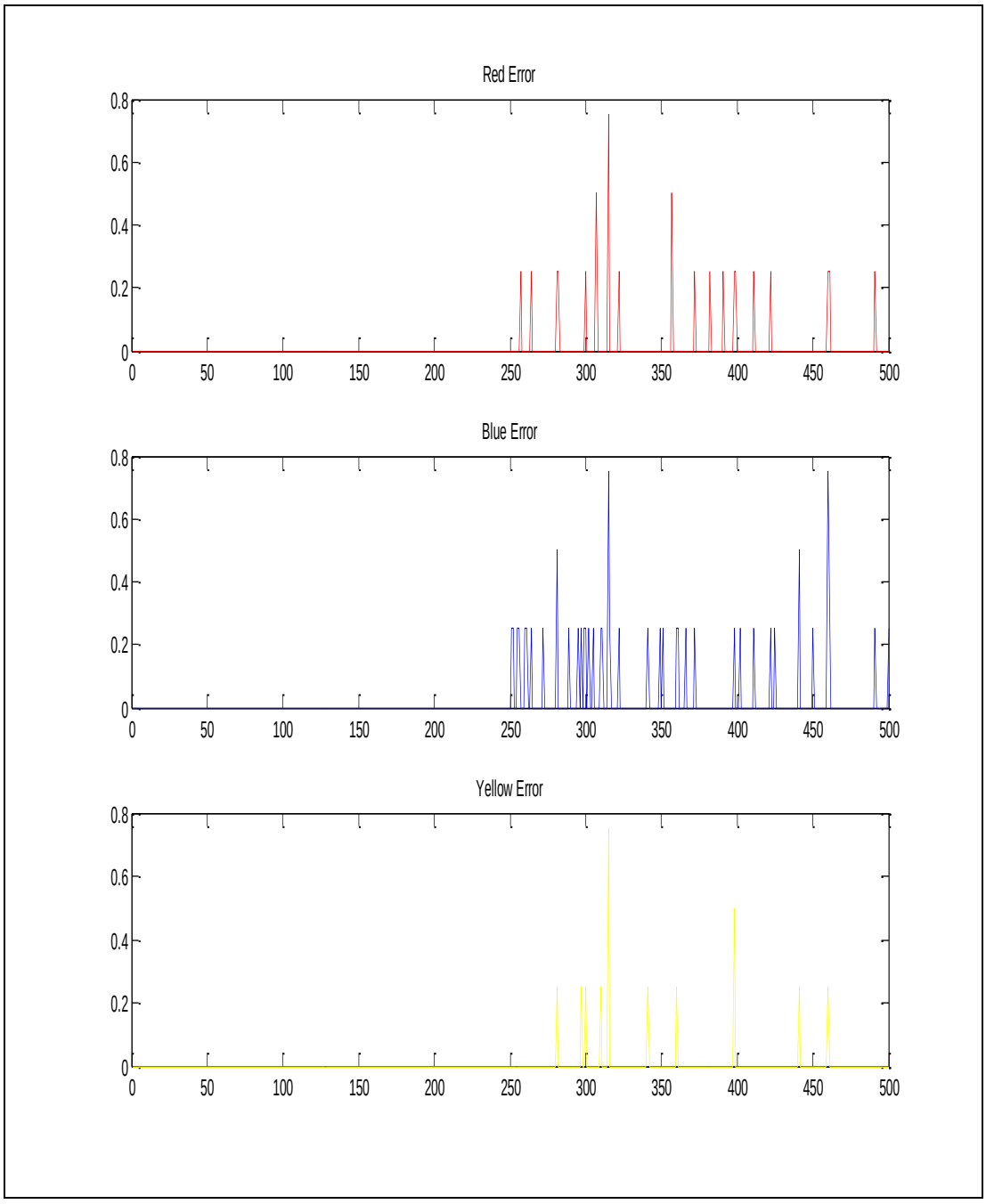

Figure 5.64 The error values for MF of 4

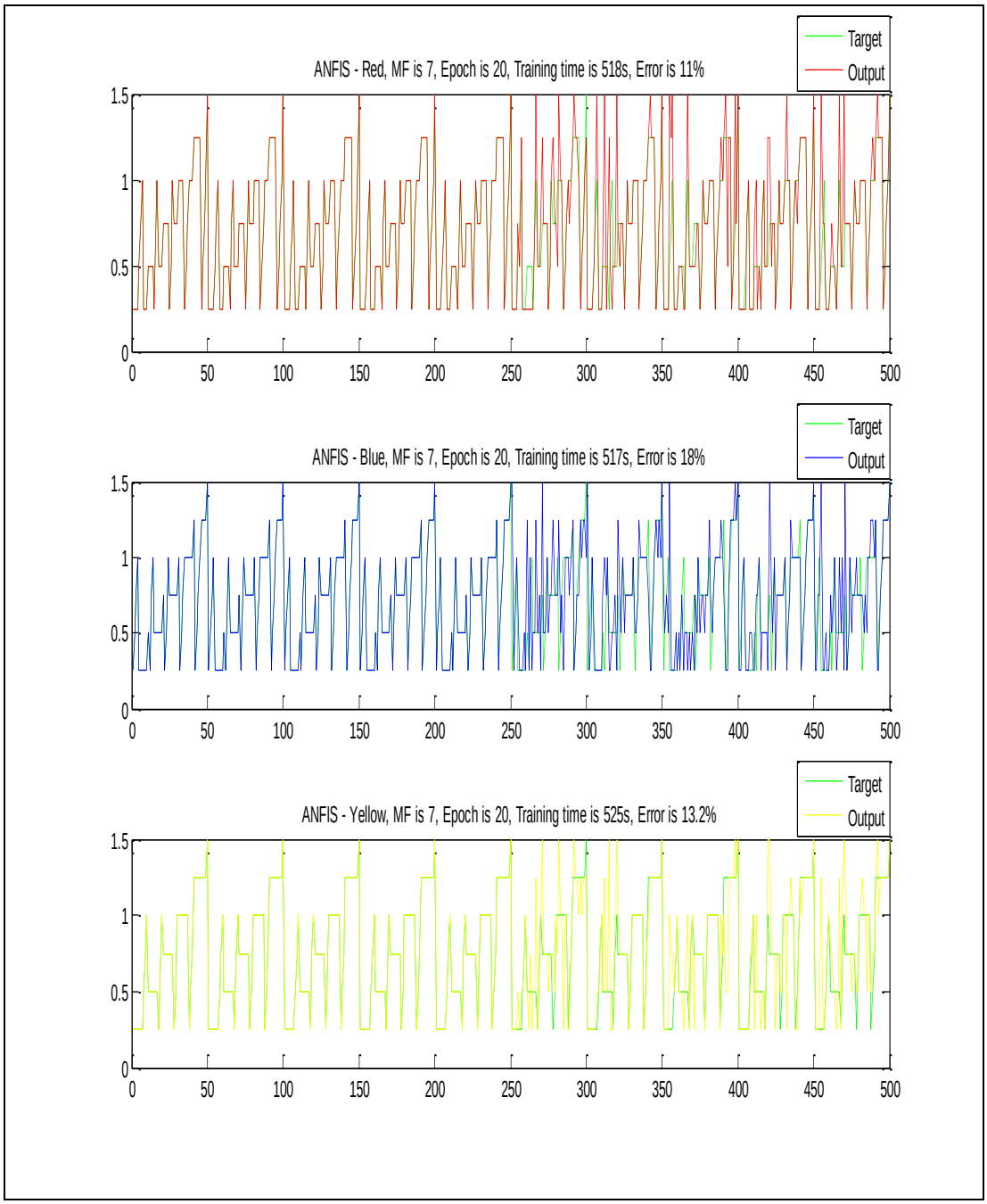

Figure 5.65 The output distribution of MF of 7 and epoch number 20.

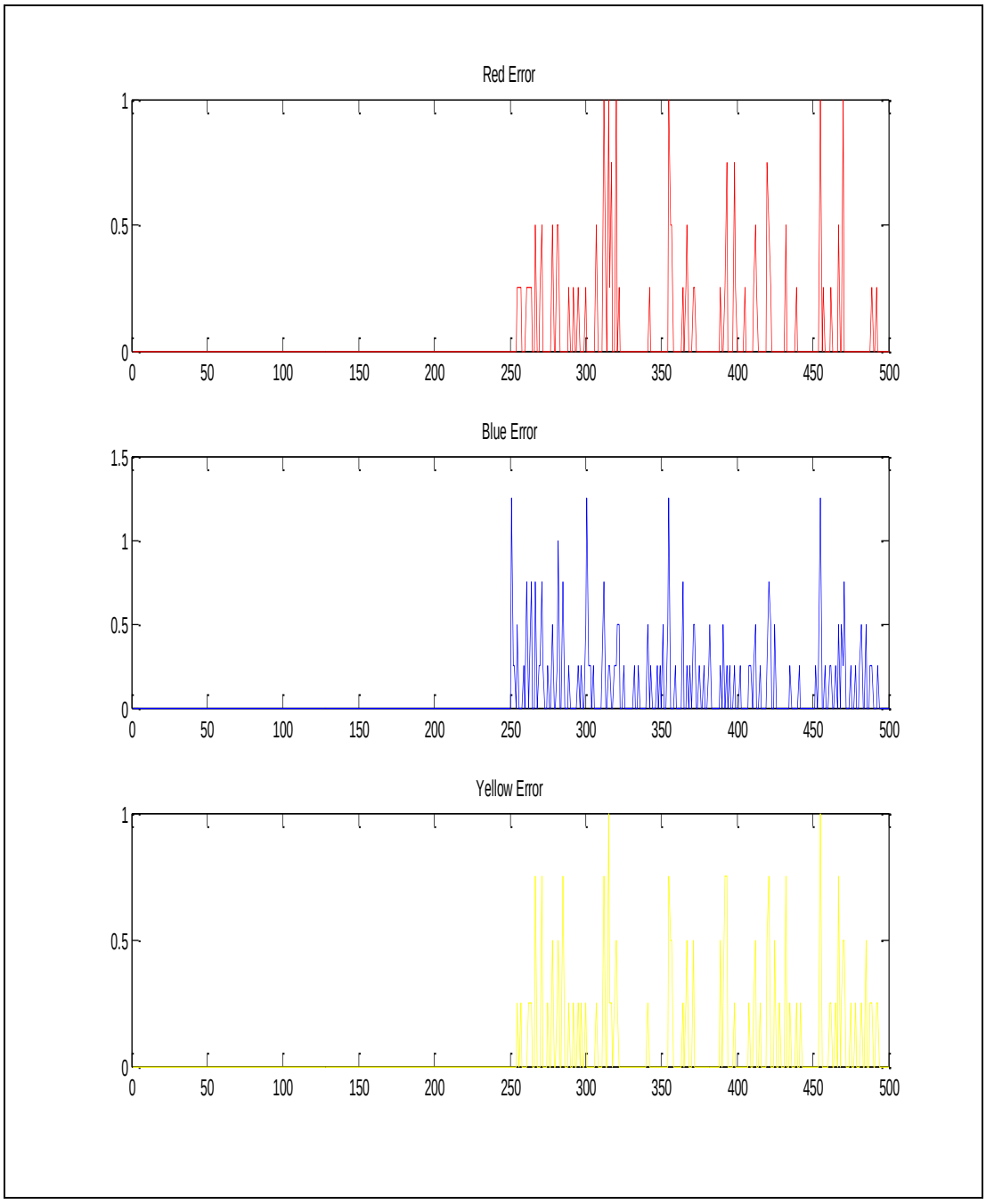

Figure 5.66 The error values for MF of 7

In the training first of all, the MF number was increased from 2 to 4. It was seen that the time of training was increased from 0-1 sec to 9 sec. However, the training performance was increased and error percentages were decreased.

Secondly, it was tried to observe that what will happen if the MF number was increased from 4 to 7. It was seen that both the training time and error percentages increased. The training time increased from 9 sec to 518 sec. However the training

performance of the system was decreased. The figures from 5.61 to 5.66 and explanations of these figures are given above for this application.

These experiments show that increasing the MF number too much does not mean that the system will have a good performance of training. It has a limitation.

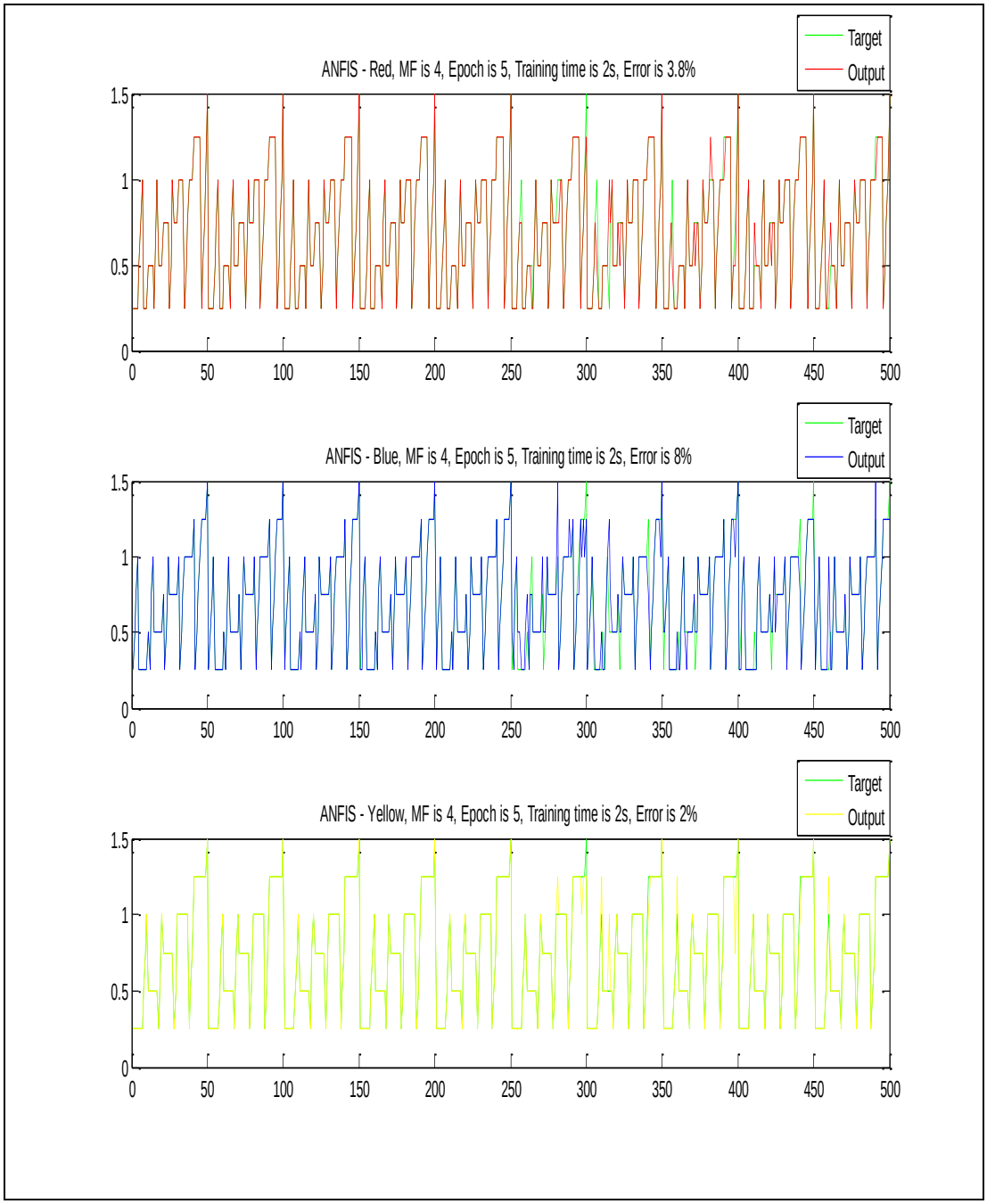

Figure 5.67 The output distribution of MF of 4 and epoch number 5.

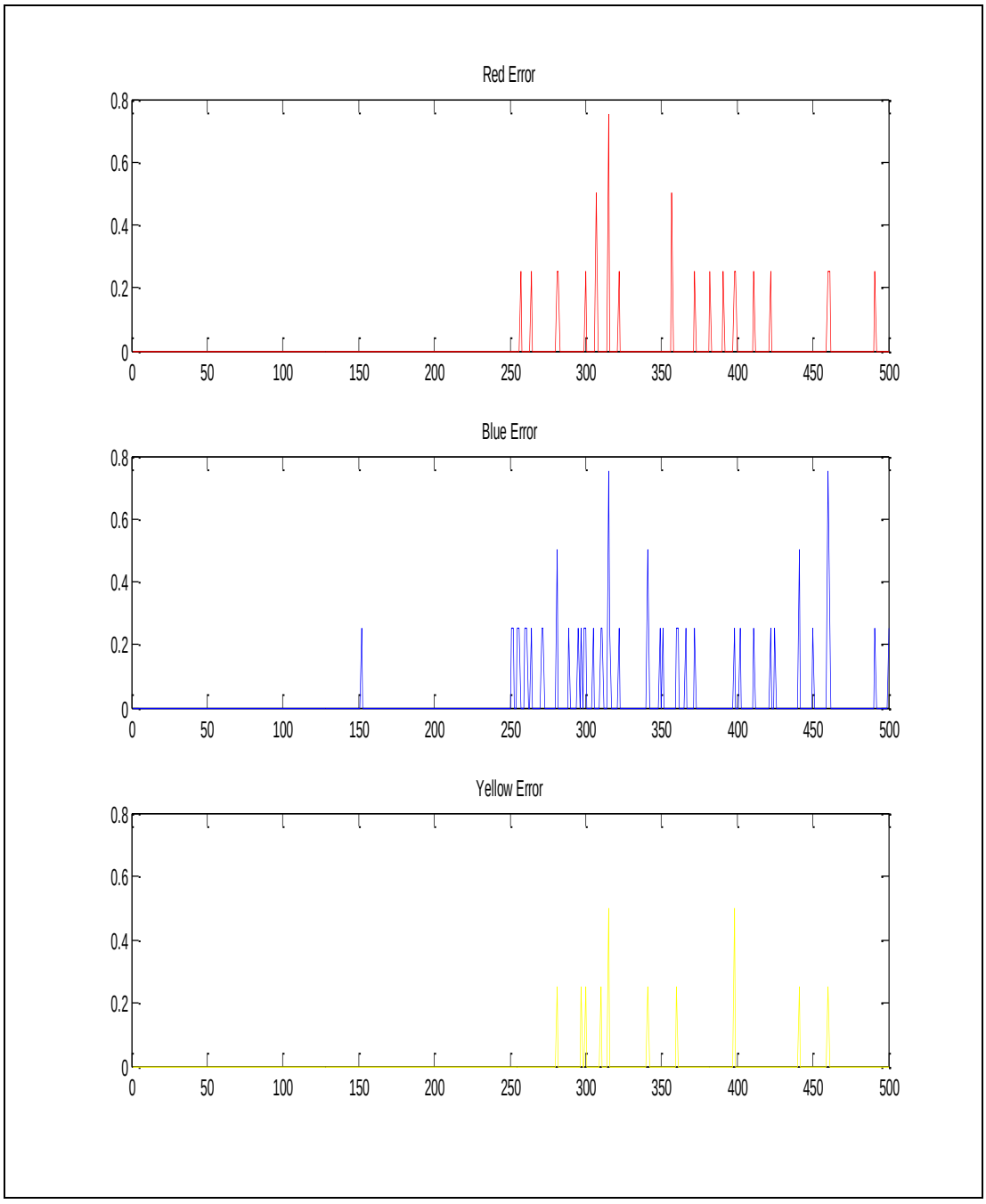

Figure 5.68 The error values for epoch number of 5

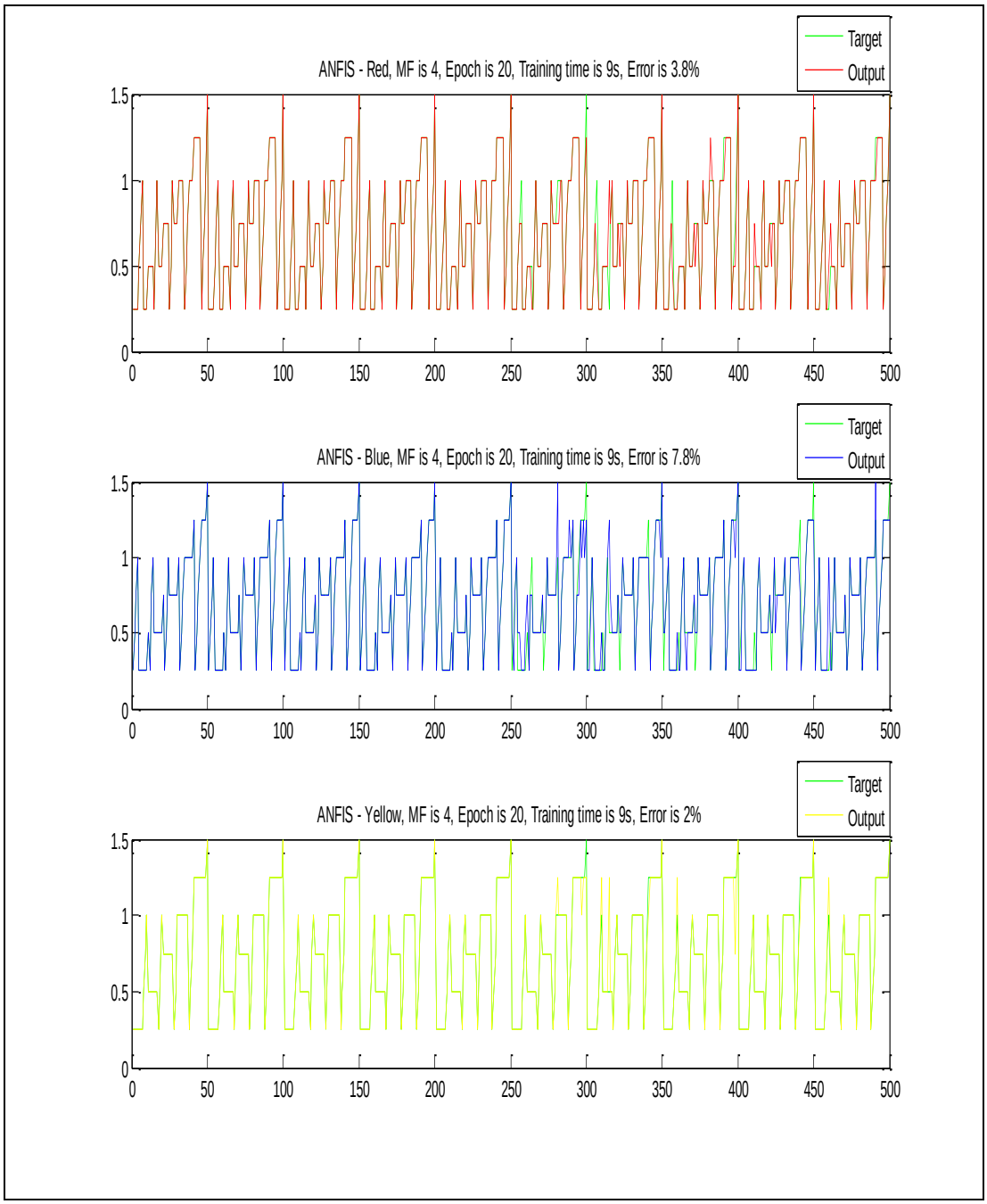

Figure 5.69 The output distribution of MF of 4 and epoch number 20.

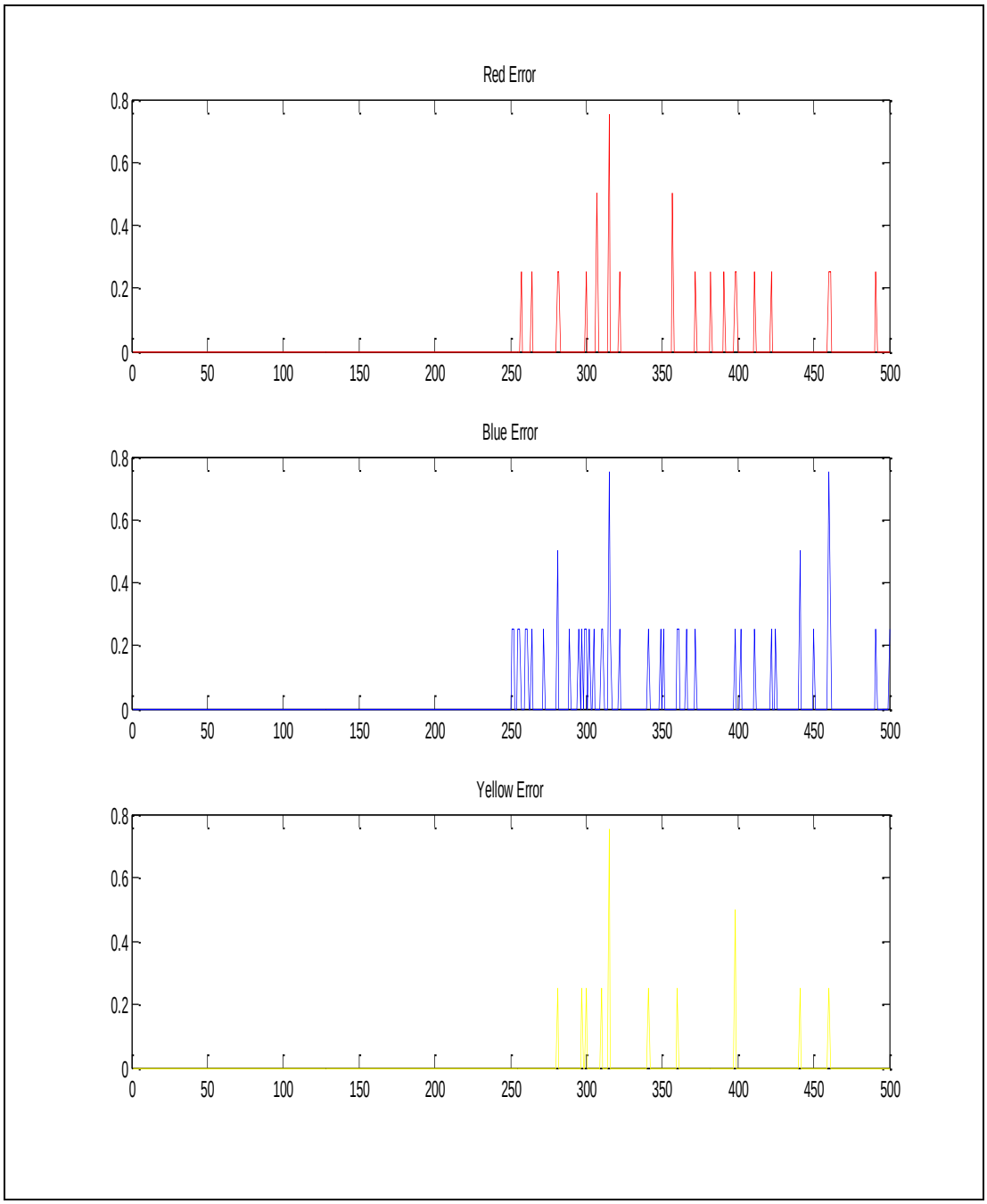

Figure 5.70 The error values for epoch number of 20

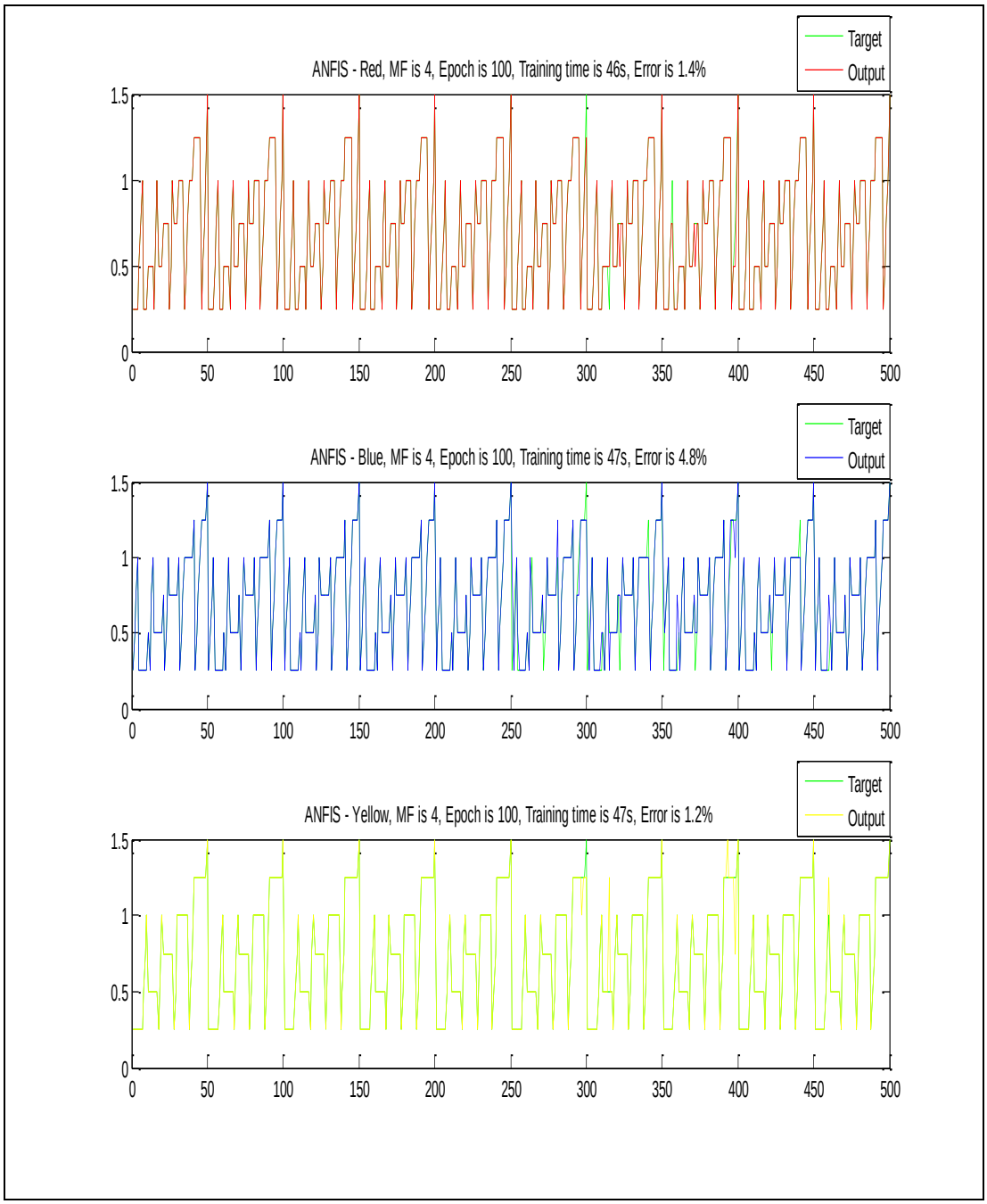

Figure 5.71 The output distribution of MF of 4 and epoch number 100.

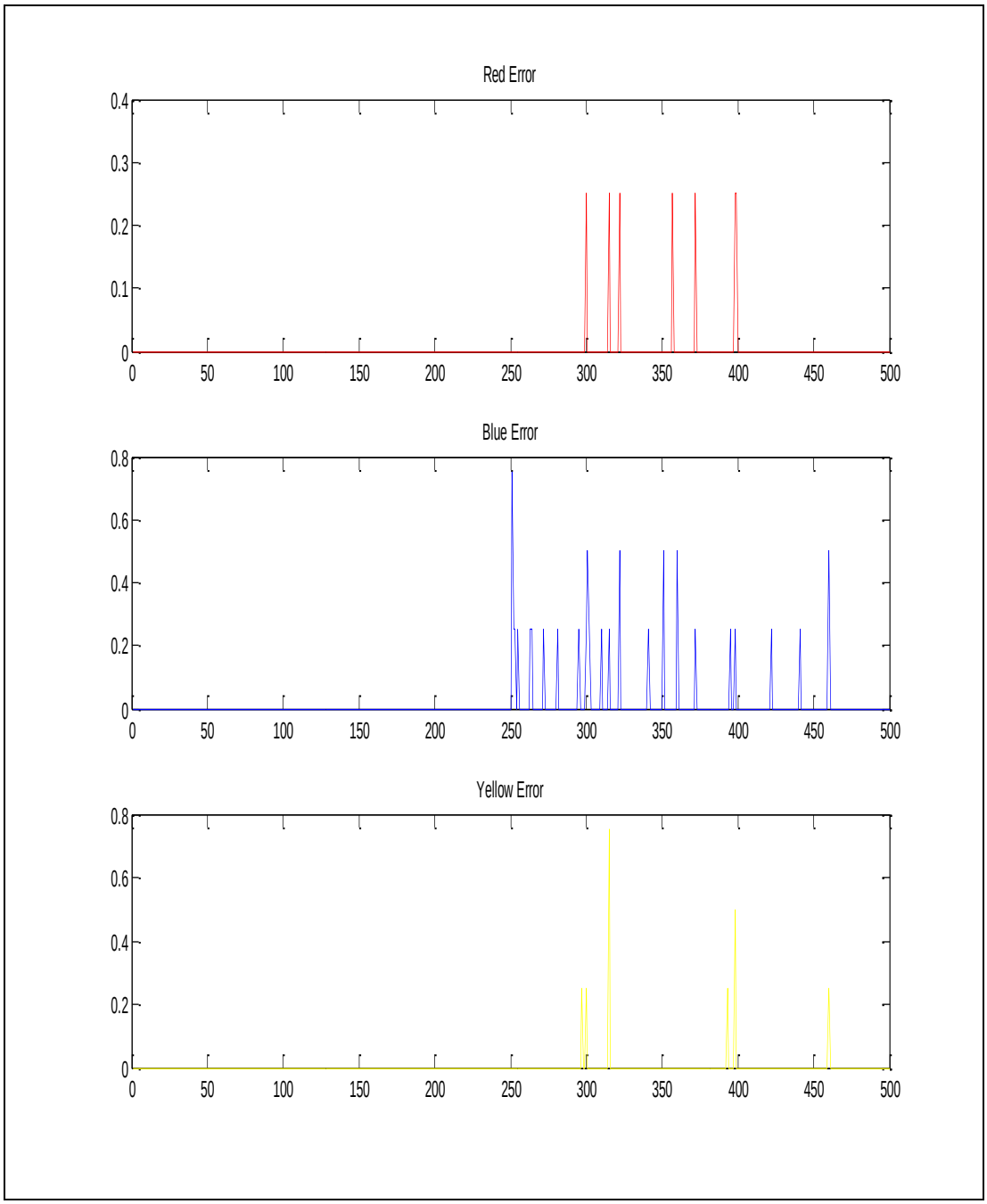

Figure 5.72 The error values for epoch number of 100

In the training first of all, the epoch number was increased from 5 to 20. It was seen that the time of training was increased from 2 sec to 9 sec. However, the training performance was increased and error percentage of blue color was decreased. Only blue color performance was changed because the other ones red and yellow error percentages were already small, thus they were not affected by the small change in the epoch number.

Secondly, it was tried to observe that what will happen if the epoch number was increased from 20 to 100. It was seen that both the training time and error percentages increased. The training time increased from 9 sec to 47 sec. And the error percentage was decreased to nearly a half.

These experiments show that increasing the epoch number too much means that the system will have a good performance of training. All of the color outputs were affected positively by this change. The figures from 5.67 to 5.72 and explanations of these figures are given below for this application.

At last, we increased the training data set number from 250 to 400. The results for 250 inputs are shown in the Figure 5.71. The results of 400 inputs are shown in Figure 5.73.

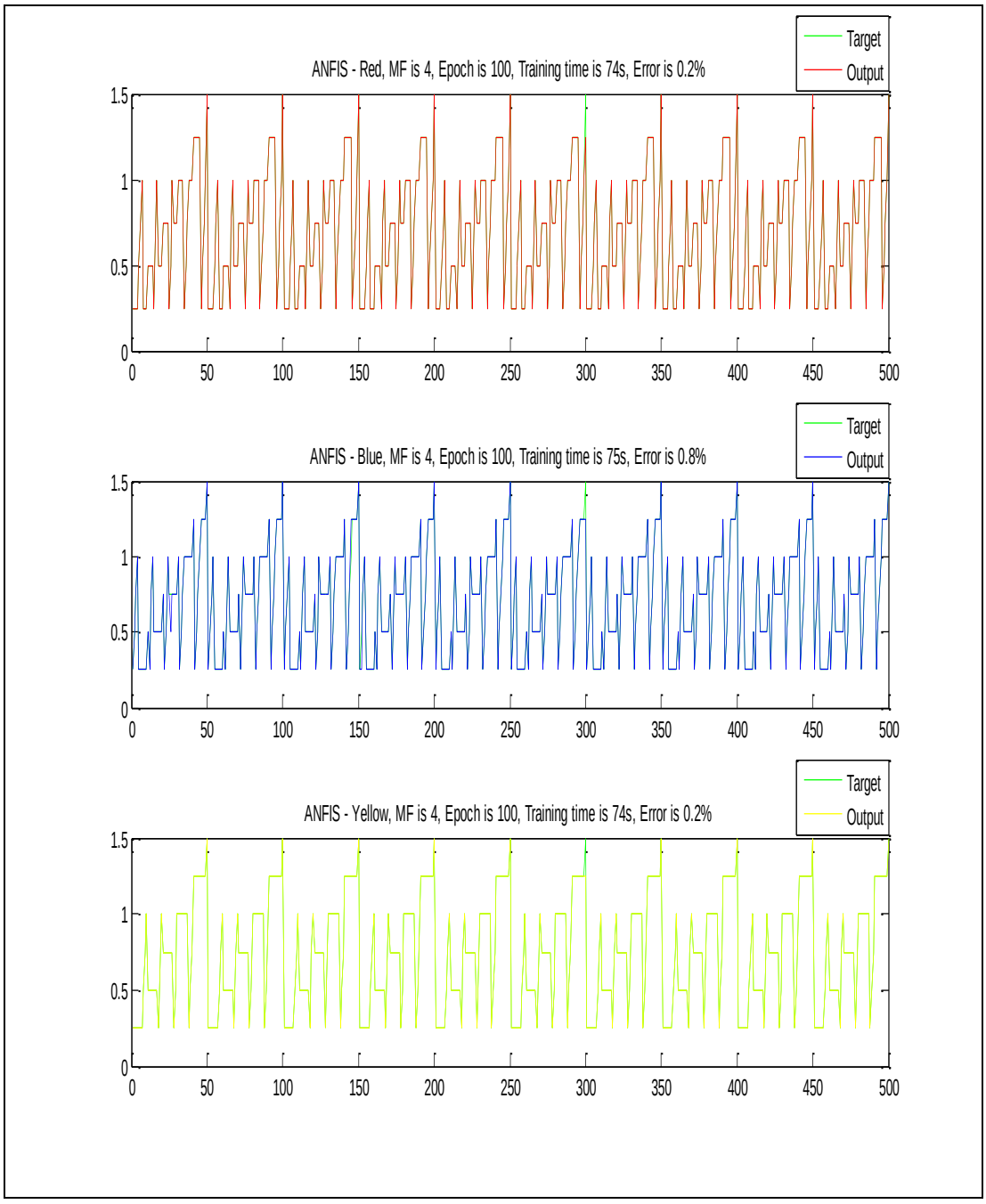

Figure 5.73 The output distribution for input number of 400

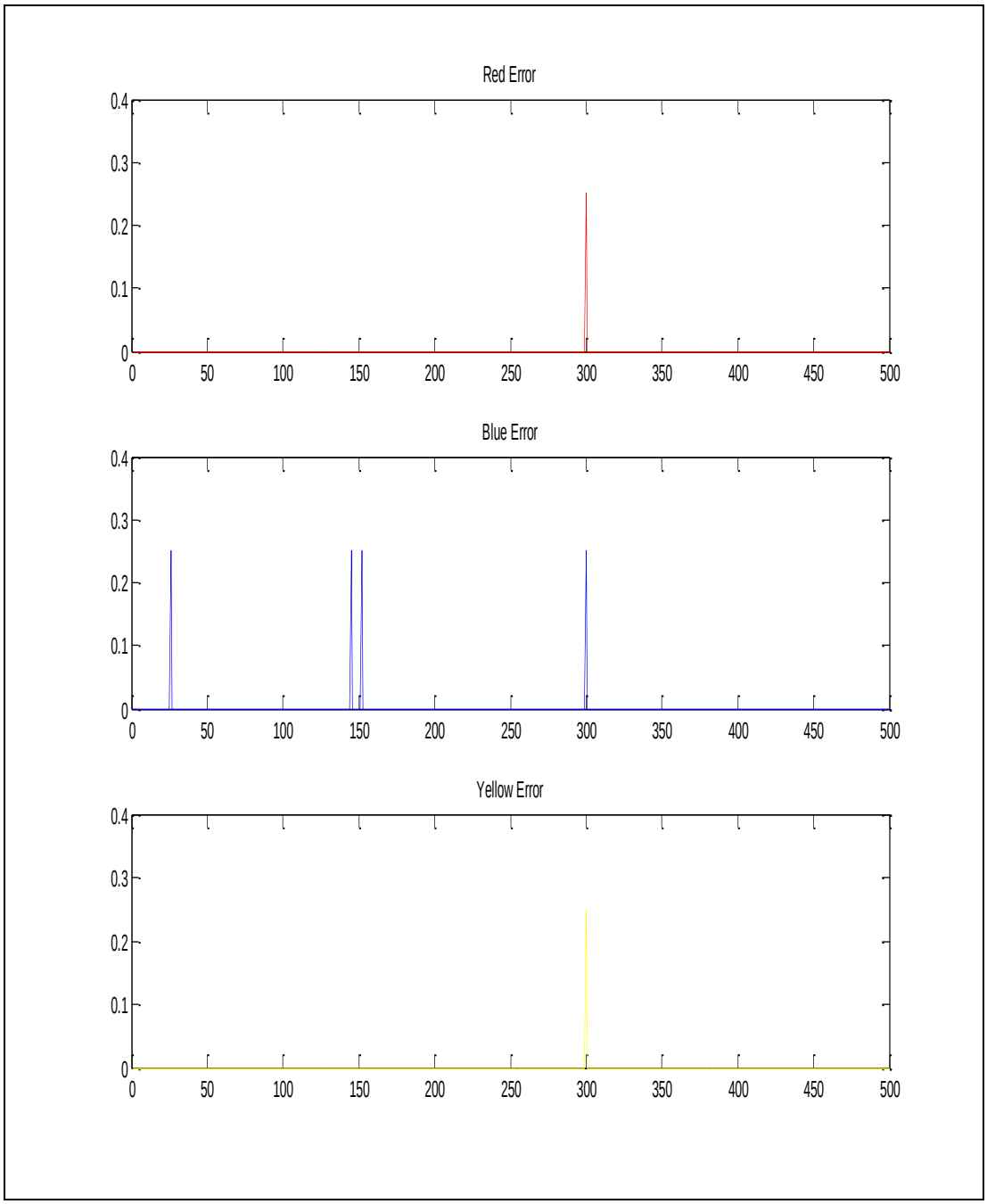

Figure 5.74 The error values for input number of 400

In this application the training data set number is increased from 250 to 400. By this way, the error values were decreased to a quite smaller value. Thus as mentioned in the conclusion part, the number of the training data is very important for almost all of the methods in order to increase the performance of the system. The figures 5.73 and 5.74 and explanations of these figures are given above for this application.

CHAPTER SIX

CONCLUSION

As it was mentioned above, color recipe prediction is a very important problem in textile industry. There are a lot of methods of prediction of color recipes. In this thesis, color recipe prediction with neural networks and fuzzy logic were applied.

In artificial neural network applications, RBF and MLP neural network were used as methods. In fuzzy logic anfis method was used. These applications have some advantages and disadvantages over the other ones. In this part, these properties will be discussed in details.

First of all, neural network methods will be compared. In RBF application one of the most important tasks is to choose the right spread value that affects the system performance. Once the spread value is chosen correctly the system is trained well. The most important advantage of RBF is its convergence speed. Although the data were too large, it had low training times. In RBF, there are some methods available in Matlab. We applied in this thesis two of the most successful methods in order to compare their performances. These methods are newrbe and newrb. In the experiments it was seen that they have similar results, however, newrbe has somehow more exact results than newrb. Newrbe is faster than newrb in the training process; also it has smaller error percentages of training and testing. Therefore with this observation, most of the applications were realized with newrbe.

Secondly, MLP was used as another neural network method. In MLP, there are also some methods of training. Newff method was chosen for the training of our system. It was chosen because it is more easy to use and it is also a more general method than the others. There were some parameters in order to adjust the training process. These parameters are hidden layer number, hidden neuron number, the goal, and the epoch number. The disadvantage of newff is the correct adjustment difficulty of these parameters at the same time. The relationship between these parameters must be

adjusted very sensibly in every training process according to the values of training data. Therefore, its adjusting and training periods are too long.

The last method used for training of the system was anfis method of fuzzy logic. In anfis, there are two parameters that affect the training of the system. These parameters are epoch number and membership function (mf) number. Increasing the mf has a positive effect on the training. The biggest disadvantage of anfis is while increasing the membership function number there are a lot of memory limitations. Because of this, it is not easy to increase the mf number to a very high value. Therefore, this parameter is not useful in this training. The other parameter that affects the system is the number of epochs. When we increase the epoch number, the system performs better. However, after a limitation it does not affect the system at all. Because of this, the epoch number was used in a limited value.

In the application of these three different methods, it was seen that RBF was faster and has more exact results than MLP and anfis. In addition, the number of adjustment parameters of RBF is less than the two others. Therefore, we say that for our purpose it is easy to implement RBF for training. Moreover, RBF has more advantages than MLP and anfis for this application. The results of MLP and anfis are mostly the same. However, the training time of MLP is less than anfis. Because of this, MLP can be preferred compared to anfis.

As a result, RBF is the most useful and successful method in this application. Although the other methods can be used, RBF is the most effective according to the results. It is more stable than the other systems with its zero error percentage and with its less training time.

In all performed applications it was clearly seen that error value decreased with increased training data set. This means, the number of the training data is very important for the training performance of the system. It can be said that the number of training samples is the most important parameter in all of the methods used. Thus, by using insufficient training data, the system cannot be trained well.

REFERENCES

Artificial neural network. (n.d.). Retrieved August 21, 2009, from http:// en.wikipedia.org/wiki/Artificial_neural_network

Bhattacharjee, D. & Kothari, V. K. (2007). A neural network system for prediction of thermal resistance of textile fabrics. *Textile Research Journal, Vol 77(1)* 4–12.

Çeven, E. K. & Özdemir, Ö. (2007). Using fuzzy logic to evaluate and predict chenille yarn shrinkage behavior. *Fibres & Textile in Eastern Europe, Vol 15* No 3(62).

Duran, K. (1997). Bilgisayarlı renk ölçüm sistemleri yardımı ile boyama reçetelerinin çıkarılması, *Tekstil ve Konfeksiyon, 2/1997,* 121-123.

Duran, K. (2001). Tekstilde renk ölçümü ve reçete çıkarma. *E.Ü. Tekstil ve Konfeksiyon Araştırma-Uygulama merkezi Yayını, Yayın No: 17.*

Ergan, Z., H., Çukul, D. & Öztürk, M. M. (2009). Utilazation of artifical neural network in textile research. *Autex 2009 World Textile Conference,* 1394-1403

Fuzzy Logic Tutorial. (n.d.). Retrieved September 11, 2009, from http:// www.seattlerobotics.org/Encoder/mar98/fuz/flindex.html

Golob, D., Zupan, J. & Osterman, D. P. (2008). *The use of artificial neural networks for color recipe prediction in textile printing*. Retrieved September 10, 2009, from http://www.autex2008.it/cd/files/24/artificial_neural.pdf.

Ham, F. M. & Kostanic, I. (2001). *Principles of neurocomputing for science and engineering*, Singapore: Mcgraw Hill International Edition.

Haykin, S. (1999). *Neural networks a comprehensive foundation* (2nd edition). United States of America: Prentice Hall International, Inc.

Kandi, S.G. & Tehran, M. A. (2007). Color recipe prediction by genetic algorithm. *Science Direct, 74,* 677-683

Nave, R. (n.d.). *The C.I.E. Color Space*. Retrieved August 11, 2009, from http:// hyperphysics.phy-astr.gsu.edu/hbase/vision/cie.html

Neural networks. (n.d.). Retrieved August 11, 2009, from http://www. learnartificialneuralnetworks.com

Passino, K. M. & Yurkovich, S. (1998). *Fuzzy control.* United States of America: Addison-Wesley Longman, Inc.

Starr, C., Evers, C. & Starr, L. (2005). *Biology: Concepts and applications* (6th edition). Thomson Brooks/Cole.

Senthilkumar, M. (2006). Modelling of CIELab values in vinyl sulphone dye application using feed-forward neural networks. *Dyes and pigments, 75*, 356-361.

Senthilkumar, M. & Selvakumar, N. (2005). Achieving expected depth of shade in reactive dye application using artificial neural network technique. *Dyes and pigments, 68,* 89-94.

Sivanandam, S. N., Sumathi, S. & Deepa, S. N. (2007). *Introduction to fuzzy logic using MATLAB.* Berlin:Springer.

The biological model: The human brain. (n.d.). Retrieved August 11, 2009, from http:// www.nnwj.de/biological-model-human-brain