DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

SPEECH RECOGNITION FOR TURKISH PHONOLOGY USING WAVELET TECHNIQUES

by
Tolga GÜLOĞLU

June, 2014

İZMİR

SPEECH RECOGNITION FOR TURKISH PHONOLOGY USING WAVELET TECHNIQUES

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
Tolga GÜLOĞLU

June, 2014

İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"SPEECH RECOGNITION FOR TURKISH PHONOLOGY USING WAVELET TECHNIQUES"** completed by **TOLGA GÜLOĞLU** under supervision of **ASST. PROF. DR. GÜLDEN KÖKTÜRK** 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.

Asst. Prof. Dr. Gülden Köktürk

Supervisor

Asst. Prof. Dr. Nalan ÖZKURT

(Jury Member)

Prof. Dr. Alp KUT

(Jury Member)

Prof.Dr. Ayşe OKUR
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

Firstly I would like to thank my supervisor Asst. Prof. Dr. Gülden KÖKTÜRK, who always supported and behaved me very kindly throughout my thesis.

I would like to thank to my company Türk Telekom A.Ş. and my administrators for giving me the opportunity to improve my thesis. Also I want to thank to my family, my mother Nermin, my father Günay and my dear friend İrem for their support. I always felt their support during my studies.

Tolga GÜLOĞLU

SPEECH RECOGNITION FOR TURKISH PHONOLOGY USING WAVELET TECHNIQUES

ABSTRACT

In this thesis, Turkish speech recognition related to phonology, recognition methods and rules of these methods are presented. Also similarities and differences between Turkish and foreign languages are shown but study based on unique features of Turkish language.

For Turkish speech recognition, firstly wavelet coefficients are obtained by wavelet transform for all recorded words. Then these coefficients are used as input data and they analyzed in artificial neural network for distinguishing words.

Although there are lots of different studies in the this subject for other languages, especially for English, there is not so much on Turkish and Turkish phonology. The book of Türkçenin Ses Dizimi by Ömer Demircan is used for source and referenced for Turkish phonology.

Keywords : Turkish phonology, speech recognition, voice recognition, wavelet technique, wavelet transform, artificial neural network, signal processing.

DALGACIK TEKNİKLERİ KULLANARAK TÜRKÇE SES BİLİMİNDEN YARARLANARAK TÜRKÇE SES TANIMA

ÖZ

Bu tezde, Türkçe'nin sesletim özellikleri kullanılarak kelimelerin söyleniş ve seslendirme açısından ayırt edilmesi, ayırt etme yöntemleri ve bu yöntemlerin hangi kurallara bağlı olarak değişiklik gösterdiği sunulmaya çalışılmıştır. Ayrıca seslendirme açısından Türkçe'nin diğer dillerle olan benzerlik ve farklılıkları da gösterilmiş, Türkçe'nin kendine has seslendirme yapısı üzerinde durulmuştur.

Türkçe kelimelerin ayırt edilmesi için öncelikle dalgacık teknikleri yardımıyla kayıt edilmiş ses dosyalarından katsayılar elde edilir. Daha sonra elde edilen bu katsayılar yapay sinir ağları yönetimi ile analiz edilerek kelimeler ayırt edilir.

Bu çalışma konusunda başta İngilizce olmak üzere yabancı diller için birçok çalışma ve araştırma olmasına rağmen Türkçe ve Türkçe sesletimi için çok fazla çalışma bulunmamaktadır. Türkçe'nin doğru sesletimi ve okunuş kuralları konusunda Ömer Demircan'ın Türkçenin Ses Dizimi adlı kitabı kaynak olarak kullanılmıştır.

Anahtar kelimeler : Türkçe ses bilimi, konuşma tanıma, ses tanıma, dalgacık tekniği, dalgacık dönüşümü, yapay sinir ağı, sinyal işleme.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE
INTRODUCTION

Speech recognition is based on distinguishing spoken words by analyzing phonological features of a language. Phonology of Turkish is the main criterion for Turkish speech recognition. There are lots of researches on phonology of Turkish, one of the most important researcher scientist on this subject is Ömer Demircan who made his first research in 1962 by comparison of Turkish and English phonology and went on his researches until 1995 with lots of different works. He has lots of studies and books which are reliable sources for Turkish phonology and acoustics today.

1.1 Aim of Thesis

Speech recognition using wavelet techniques is a very famous and popular subject recently. Many studies on this subject is based on English speech and voice recognition. This thesis is based on Turkish speech recognition by using criterions of phonology of Turkish.

Wavelet transform is used for converting voice samples into series of numeric values which are characteristic for every word. Some of studies on the same subject use Fourier transforms techniques for this operation. This thesis is an analysis for speech recognition performance of wavelet transform especially. For this aim, different wavelets will be used respectively in wavelet transform and their performance will be compared.

Wavelet transform techniques can be used for different purposes. There are lots of similar previous studies using wavelet techniques such as speaker independent speech emotion recognition, auditory perception, approach for speaker identification, classification objects by sound.

1.2 Outline of Thesis

The first chapter is introduction and includes main idea of thesis and basic information for phonology of Turkish. The second chapter includes information about feature extraction method, wavelet and wavelet transform. The third chapter gives information about neural network anal analysis. Chapter four includes information about development of software used for thesis. Chapter five gives experimental results and finally chapter six is conclusion.

1.3 Phonology of Turkish

Phonology is science, investigates relations and details of voice and the words for a specific language. It analysis syllables, voice groups, word particles and analysis relations between them. Also it focuses on changes of voices and words in time. A good voice change example in Turkish is change of "t" and "d" as shown.

Old : tamar , temir, töşek , tilek
New : damar , demir, döşek , dilek

Phonetics is science, researches on nature features of voices in a specific language. As an example, at this viewpoint there are lots of different "k" voices.

- With regard to touch location of tongue and palate: kim, kes, kıy, kal.
- Differentiates from above words by globosity of lips: küt, köy, kum, kon.
- With regard to aspiration: dik, sek, sık, bak.

As seen above examples, letter "k" has twelve different voices. New examples can be adduced so it's clear that letter "k" has lots of different voices. All these different "k" voices are distinct and unique voices, so related to usage, all these Turkish "k" are different members of Turkish letter voice set.

Phonology is science that investigates functions of voices in a specific language. This science analysis both pronunciation and voices of the language.

Phonetics ⟷ **Phonology** ⟷ **Morphology**

Phonology has a strong relation with evolution of writing. In the beginning, writing was a kind of drawing including figures and it has no relation with voice. By the time shapes and figures became syllables which have reference to voices. Today writing is based on letters which are stand for voice pieces and named as “abece”.

When “abece” is system discovered, workings of writing are started by the nations. Unity orthography was the common mark but some conflicts occurred. A very good examples of these conflicts is some writing differences between American English and English English. For example, although “color” and “colour” have the same spelling, have different writing in American English and English English. Also Ottoman language writing has lost of these kind of inconsistencies.

To prevent Turkish from these inconsistencies, Türk Dil Kurumu releases a book in the name of Yazım Klavuzu which has the rules of both writing and spelling. But in spite of protection, after 1980s, serious inconsistencies were occurred in Turkish.

By the years, it’s understood that voice is not only related to writings rules, also it can be related to stress and emphasis in some languages. For example, in the sentence of “Did children give FLOwers to the teacher?”, the stress is based on flowers. Because in English, the emphasis of sentence is provided by pronunciation. Unlikely to English, in Turkish, emphasis of sentence is provided by location of words as seen in the examples in below;

“Çocuklar öğretmene çiçek Mi verdi?” => Emphasis of sentence is flowers.
“Çocuklar öğretmene çiçek verdi Mİ?” => Emphasis of sentence is giving or not.
“Çocuklar MI öğretmene çiçek verdi?” => Emphasis of sentence is children
“Çocuklar öğretmene Mİ çiçek verdi?” => Emphasis of sentence is teacher.

Turkish has also its own pronunciation rules. Lost of studies are made on this issue between 1920's and 1980's, which are based on "İstanbul Türkçesi" and finally collected in Yazım Klavuzu by Türk Dil Kurumu. In other words, Yazım Klavuzu shows and explains the rules of writing of a word related to grapheme but not includes emphasis. Only the foreign-funded words don't obey Turkish spelling and writing rules. For example;

Utterance : /fi'lim / , /ti'ren / , /tıra'fik /

Writing : film , tren , trafik

1.4 Creation of Voice

Voice is produced by the vocal organs of human. The primary mission of vocal organs are completely different form creating the voice. For example, the primary missions of lungs are respiratory and cleaning the blood, the primary missions of tongue are sense of taste and swallow. Creating voice is the secondary mission of these organs. Vocal organs separated to moving and motionless organs.

1.4.1 Moving Vocal Organs

Moving vocal organs of human are lungs, plica vocalis, tongue, velum and lips. Lungs are provides necessary air to create sound. Plica vocalis is a kind of muscular that is placed at end of the windpipe. Its stretching and expansiveness creates and effect to voice directly. Tongue is the most flexible organ in vocal organs and it work with palate. Its shape tones the voice. Velum is the most mellow part of palate. It controls air flow by covering or uncovering the nasal passage. And lip is the last organ in the voice way. Its shape and position determines the characteristic of the voice.

1.4.2 Static Vocal Organs

Static vocal organs are hard palate and teeth. Hard palate is tough part of the palate. It works with front side of tongue and adjusts amount of air flow. Teeth are separated to two parts in the mouth, these are top teeth and bottom teeth. In creating the voice, only the top teeth are used. They determine the voice characteristic by the help of tongue.

1.5 Grapheme

In Turkish and other languages, there some main significant features of vocalization. These are syllable structures and duration, emphasis and tone of voice.

1.5.1 Syllable

Respiratory can’t go on continuously while speaking. For this, voices are come out part by part. These parts are named as Syllable.

Related to way that air comes out, air way has a disabled or unhindered structure. If it is disabled way then because of closing and opening of the way, voice has a sudden end. These kind of voices are named as consonant, for example p,t,f,s… etc. If air way is unhindered, then the voice can go on until the inside air finishes. These kind of voices are named as vowel, example a,e,u,i…etc.

Syllable is a kind of combination of vowel and consonant voices. In a syllable, vowel voice is imperative so vowel is named also carrier of syllable. Related to positions of vowel and consonant in a syllable, syllable has two kinds of type.

The first type is opened syllable, which starts with a consonant and ends with vowel or just has a vowel. For example; O, su, de, ne, şu.

The second type is closed syllable, which starts with a vowel and ends with consonant, or starting with a consonant and goes on with a vowel and ends with a consonant. For example; Of, ah, ey, ben, sis, düş, gel.

1.5.2 Distinguish Features of Voice

All the words has specific voice feature that makes it different from other words. Mankind can distinguish voices related to these features. These main features are duration, emphasis and tone.

1.5.2.1 Duration

This feature is based on the duration of vowel in a syllable. For example, in the words of dağ, yağ, çiğ, the letter “ğ” makes the vowel longer. Unlikely these examples, in ek, dök, ak, vowel is short and so the syllable is short. This is a specific feature to distinguish syllables and voices.

1.5.2.2 Emphasis

Emphasis is created by pressure of breath and its timing. It’s separated to two types, weak and strong emphasis. Significance of emphasis is shown in Table 1.1.

Table 1.1 Emphasis classification of example words

Word	**Emphasis Syllable**	**Mean**
Açma	ma	Patty
Açma	aç	Don’t open

1.5.2.3 Tone

Tone is a kind of sonic cover, it has descents and risings over the voice. This changing can be descent, rising, descent and then rising, rising and then descent, flat.

In Turkish, tone is usually used in the level of words. Example tone situations are showed in Table 1.2.

Table 1.2 Tone classification of example sentences and answers

Sentence	Answer	Tone
- Bunu kim yaptı?	- Ben.	Descent
- Sen çocuğa bakacaksın.	- Ben?	Rising
- Onu buraya kim çağırdı?	- Ben.	Descent and rising
- Çocuğa kim bakacak?	- Sen.	Rising and descent
- Evi kim temizleyecek?	- Sen.	Flat

CHAPTER TWO
FEATURE EXTRACTION METHOD

In this work, data obtained by wavelet transform are used as feature extraction method to analysis voice signals.

2.1 Wavelet Transform

Wavelet transform is a kind of signal transform which transforms the signal and represents it in a more useful form. Mathematically, wavelet transforms is a correlation of wavelet function and the signal.

Wavelet transform is useful for analysis the signals simultaneously in both frequency and time domains. Correlation process is computed at various locations of the signal and for various forms of wavelet. Wavelet transforms are classified into continuous wavelet transforms (CWT) and discrete wavelet transform (DWT).

Wavelet transform has a wide application area in sciences and engineering. After 1990’s, Wavelet transform became popular with a growing interest year by year. In the last years, lots of articles and journal papers concerning the wavelet transform. Wavelet transform has now been used to analysis of financial indices, hearth monitoring, air condition, seismic signals, astronomical images, video image compression, medical signal records.

2.1.1 Continuous Wavelet Transform

Wavelet transform is based on the correlation of wavelet function and the signal, this correlation is computed at various locations of the signal and for various scales of wavelet. If this correlation is done in a smooth continuous way then this correlation is named as continuous wavelet transform. In order to understand continuous wavelet transform, firstly wavelet notion should be understood.

2.1.1.1 The Wavelet

Wavelet is a function $\varphi(t)$ which satisfies a certain mathematical criteria. Basiclly, it is a localized waveform. There are lots of different wavelets, selection of wavelets depends on both nature of the signal and request of the analysis. Very commonly used wavelets are shown below.

Table 2.1 Classification of wavelets

Wavelet Family Short Name	**Wavelet Family Name**
'bior'	Biorthogonal wavelets
‘cmor’	Complex Morlet wavelets
‘cgau’	Complex Gaussian wavelets
'coif'	Coiflets wavelets
'db'	Daubechies wavelets
‘dmey’	Discrete approximation of Meyer wavelet
‘fbsp’	Frequency B-Spline wavelets
'gaus'	Gaussian wavelets
'haar'	Haar wavelet
'mexh'	Mexican hat wavelet
'meyr'	Meyer wavelet
'morl'	Morlet wavelet
‘rbio’	Reverse biorthogonal wavelets
'shan'	Shannon wavelets
'sym'	Symlet wavelets

Very commonly used and known as Mexican hat wavelet can be described as equation (2.1)

$$\varphi(t) = (1 - t^2)e^{-t^2/2} \tag{2.1}$$

In fact, Mexican hat wavelet is the negative of the second derivative of the Gaussian distribution function $e^{-t^2/2}$. All derivatives of Gaussian function may be used as wavelet. Depends on the application, the most appropriate one is used.

The first derivative of Gaussian function is named as Gaussian wavelet and it is showed in Figure 2.1. The second derivative of Gaussian function is named as Mexican hat wavelet and it is showed in Figure 2.2.

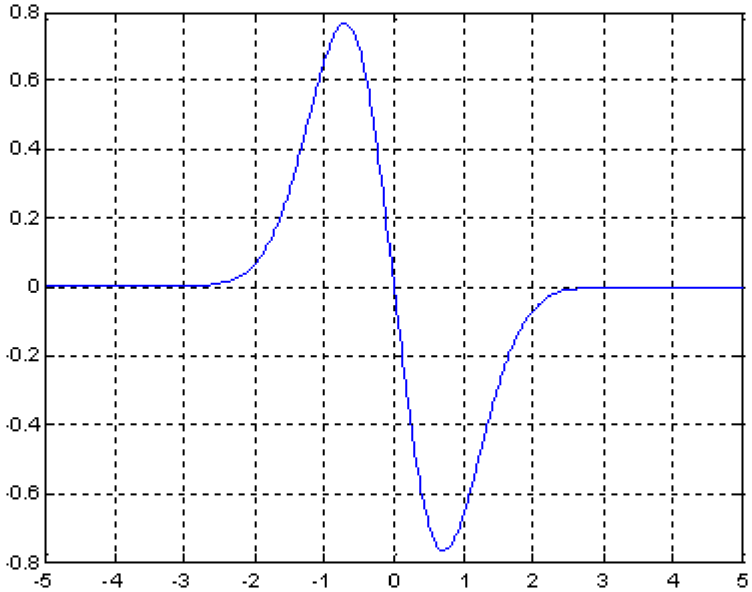

Figure 2.1 Gaussian wavelet.

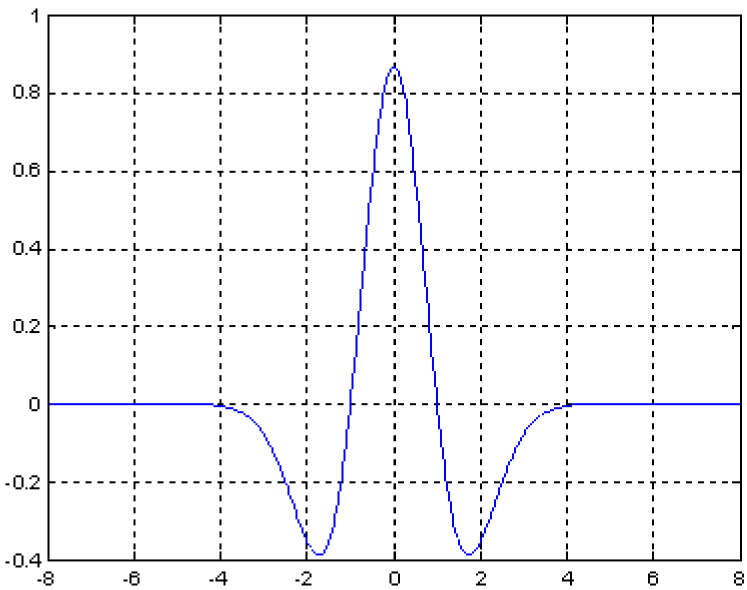

Figure 2.2 Mexican hat wavelet.

2.1.1.2 Requirements for the Wavelet

1. A wavelet must have finite energy.

$$E = \int_{-\infty}^{\infty} |\varphi(t)|^2 dt < \infty \tag{2.2}$$

where E is the energy of function.

2. If $\hat{\varphi}(f)$ is the Fourier transform of $\varphi(t)$, following condition must be hold.

$$C_g = \int_0^{\infty} \frac{|\hat{\varphi}(f)|^2}{f} \, df < \infty \tag{2.3}$$

This means that the wavelet has no zero frequency component. Equation (2.3) is also known as admissibility condition and C_g is admissibility constant.

3. For complex wavelets, Fourier transform must both real and vanish for negative frequencies.

2.1.1.3 The wavelet transform

Wavelet transform performance is based on the wavelet used and its characteristic. A wavelet is a flexible function that can be manipulated in two ways. First manipulation is dilation, it means wavelet signal can be stretched or squeezed. Second manipulation is translation or moving of the wavelet signal. Dilation and translation of wavelet signal is showed in Figure 2.3 and Figure 2.4 respectively.

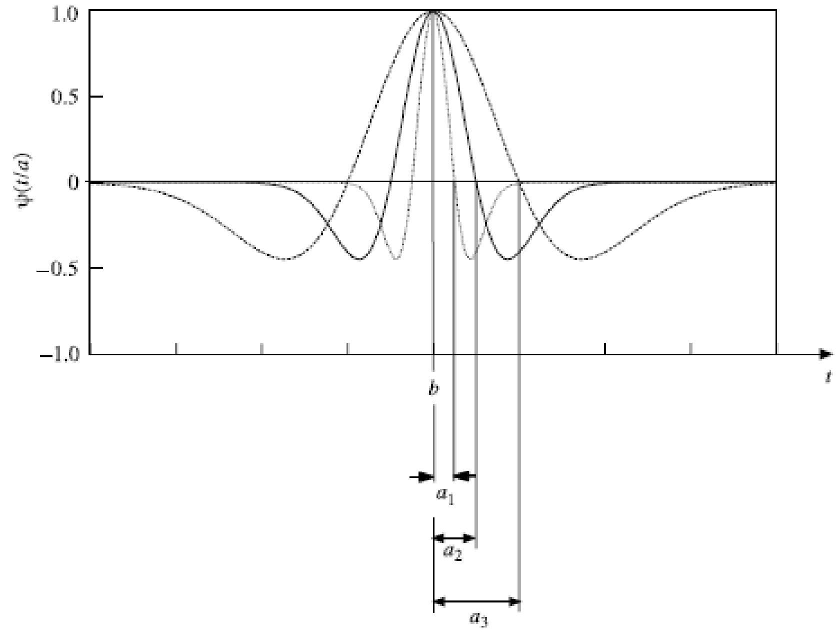

Figure 2.3 Dilation and contraction of a Mexican hat wavelet. ɑα_1 = ɑα_2 / 2 and αɑ$_3$ = αɑ$_2$ x 2.

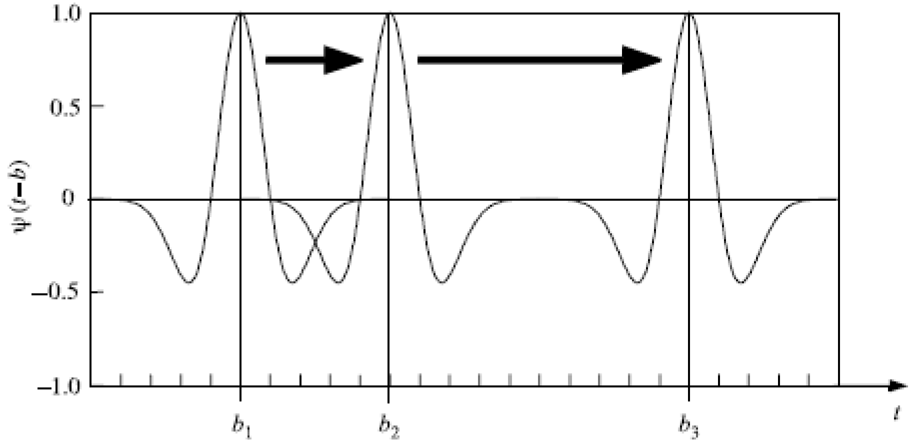

Figure 2.4 Moving of a Mexican hat wavelet.

In Figure 2.3, dilation and contraction of wavelet obtained by parameter α, which shows the distance between centre of wavelet and its crossing of the time axis. The movement of wavelet on time axis is obtained by parameter b, which shows the location of wavelet on the time axis. In Figure 2.4, Mexican hat wavelet can be seen on the different locations b_1, b_2 and b_3 along the time axis. If we use the dilation parameter α, and the location parameter b, with in our equation (2.1), we obtain shifted and dilated version of Mexican hat wavelet equation. It can be described as equation (2.4).

$$\varphi\,\frac{(t-b)}{a} = \left[1 - \left(\frac{(t-b)}{a}\right)^2\right] e^{-\frac{1}{2}[(t-b)/a]^2} \tag{2.4}$$

If a = 1 and b = 0 are given as values in equation (2.4) then equation (2.1) is obtained again as the original mother wavelet. Now, a signal *x(t)* can be transformed by using the range of a and b parameters. The wavelet transform of a continuous signal obtained from wavelet function is showed in Figure 2.5 where *w(a)* is weight function.

$$T(a,b) = w(a)\int_{-\infty}^{\infty} x(t)\varphi^*\left(\frac{(t-b)}{a}\right)dt \tag{2.5}$$

Weight function w(a) can be set to $1/\sqrt{a}$ for reasons of energy conservation. Thus wavelet transform can be written as equation (2.6).

$$T(a,b) = \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} x(t)\varphi^*\left(\frac{(t-b)}{a}\right)dt \tag{2.6}$$

Equation (2.6) is simply named as Continuous wavelet transform or CWT. This equation includes both dilated and translated wavelet $\varphi[(t-a)/b]$ and the signal *x(t).* In the equation (2.6), product of the wavelet and the signal are integrated over the signal range. In mathematical terms, this calculation is named as correlation. The normalized wavelet function van be written as equation (2.7).

$$\varphi_t(t) = \frac{1}{\sqrt{a}}\varphi\left(\frac{t-a}{b}\right) \tag{2.7}$$

Hence, the transform integral can written as equation (2.8) and simply expressed in more compact form in equation (2.9).

$$T(a,b) = \int_{-\infty}^{\infty} x(t)\varphi_{a,b}^*(t)\,dt \tag{2.8}$$

$$T(a,b) = \langle x, \varphi_{a,b}\rangle \tag{2.9}$$

2.1.1.4 Identification of Coherent Structures

Figure 2.5 shows the operation of wavelet transform by the given equation (2.8). In the figure, a wavelet located at *b* and scaled *α* is shown superimposed on a signal. Five sections are separated by the lines and sections are named as D, B, C, A and E respectively. In a region, if both of signal and wavelet are positive or negative at the same time, it has a result in positive contribution to the integral of equation (2.8). If signal and wavelet are opposite sign, it has a result in negative contribution to the integral of equation (2.8). Contribution results are showed on the Table 2.2.

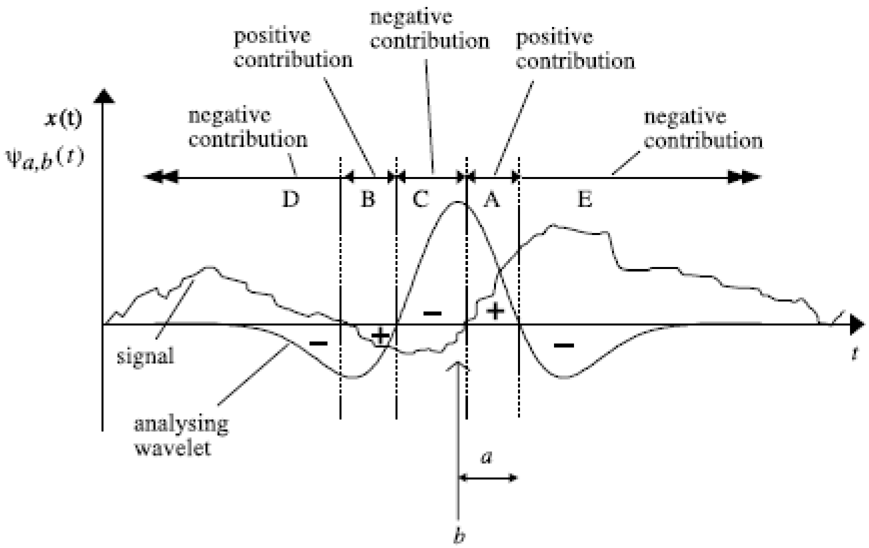

Figure 2.5 Mechanics of wavelet transform.

Table 2.2 Contribution results of regions

Region	**Sign of Signal**	**Sign of Wavelet**	**Contribution**
A	Positive	Positive	Positive
B	Negative	Negative	Positive
C	Negative	Positive	Negative
D	Positive	Negative	Negative
E	Positive	Negative	Negative

In Figure 2.6 shows a fixed scaled wavelet at four different locations on a signal. When wavelet transform is computed at these locations, four different T(a,b) values are obtained related to match of signal wavelet. For example, at b_1, a large positive value is gained, but at b_3, a large negative value is gained.

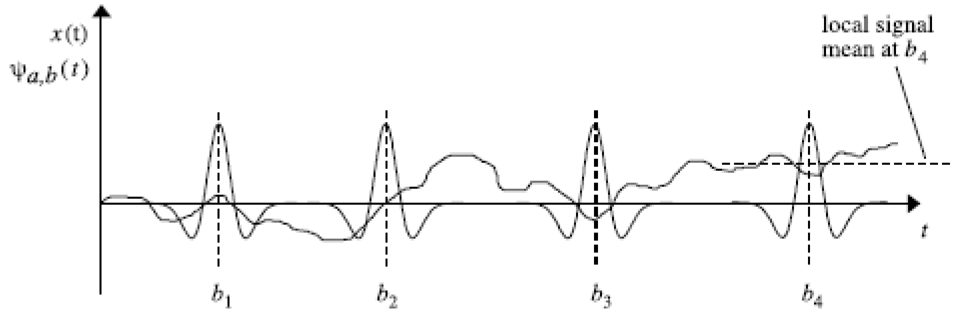

Figure 2.6 Wavelets are located four different points on a signal.

Not only the location, dilation of wavelet is also effects the result of wavelet transform. In the following figures, a fixed sinusoidal waveform and Mexican hat wavelets of various dilations are matched at different locations.

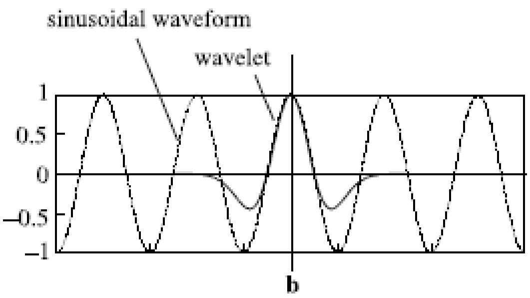

Figure 2.7 Wavelet is in phase with waveform, good positive correlation.

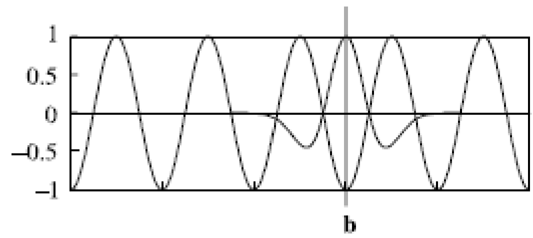

Figure 2.8 Wavelet is out of phase with waveform, good negative correlation.

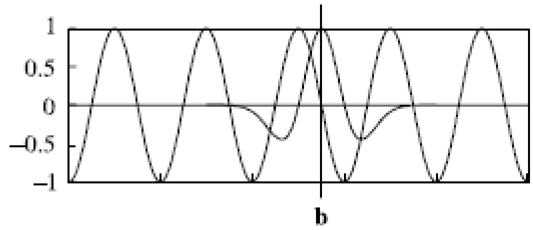

Figure 2.9 Wavelet is out of phase with waveform, zero correlation.

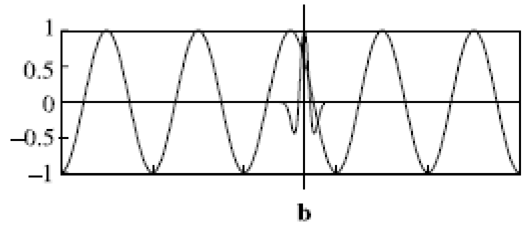

Figure 2.10 Squeezed wavelet does not match with waveform well.

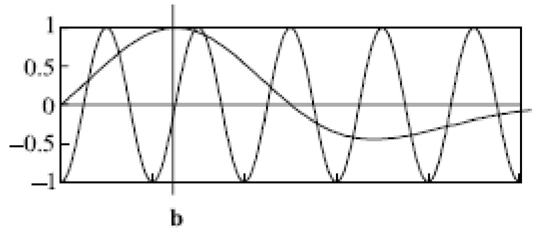

Figure 2.11 Stretched wavelet does not match with waveform well.

In continuous wavelet transform, not usually arbitrary values of dilation and location are used, mostly change of a continuous range of dilation and location are used. A plot of *T(a,b)* versus *a* and *b* for a sinusoidal signal, where a Mexican hat wavelet is used is showed in Figure 2.12. Two different methods are used to result of wavelet transform. The first one contour plot which showed in Figure 2.13 and the second one surface plot which showed in Figure 2.14.

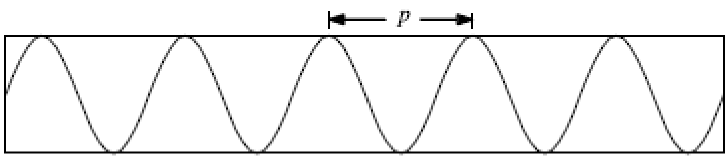

Figure 2.12 A sinusoid of period *p*

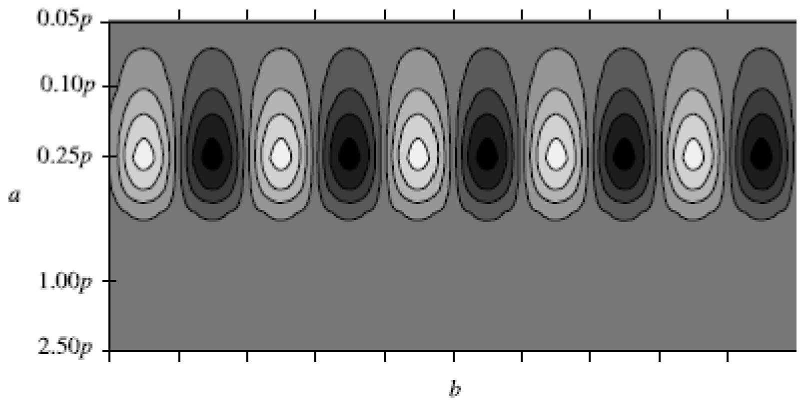

Figure 2.13 Contour plot of T(a,b) for the sinusoid in Figure 2.12.

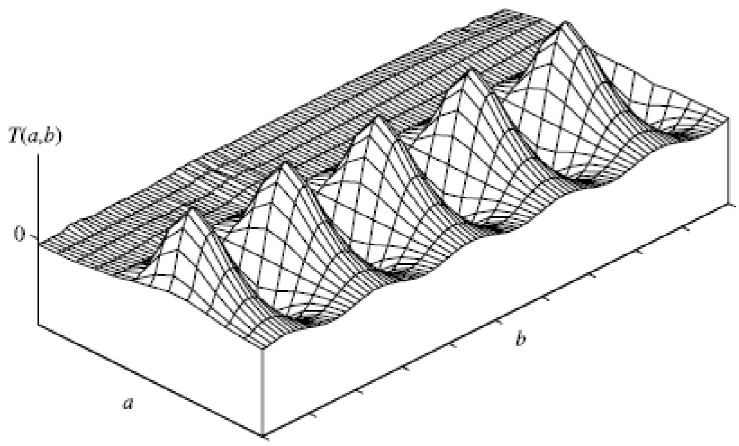

Figure 2.14 Isometric surface plot of T(a,b) for the sinusoid in Figure 2.12.

The near-zero values of T(a,b) are obtained in both small and large values of a. But at intermediate values of a, large undulations in T(a,b) are obtained due to well matching of Mexican hat wavelet and the sinusoid.

In Figure 2.15, a waveform which is composed of combination of two sinusoidal waveforms is showed. The first signal has a period of p_1 and the other signal has period of p_2, where p_1 is five times of p_2.

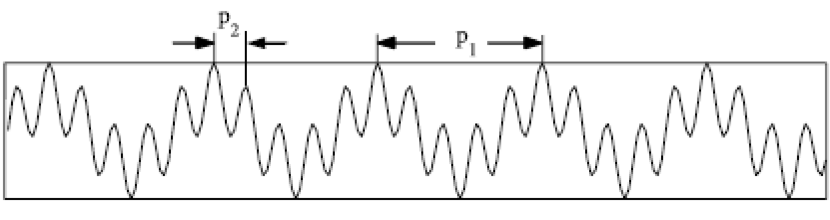

Figure 2.15 Composed of two sinusoids period of p_1 and p_2, where $p_1 = 2\ x\ p_2$.

Plot results of T(a,b) for the signal in Figure 2.13 are showed in Figure 2.14 and Figure 2.15. Plot shows two periodic waveforms in the signal. This figure shows us the ability of wavelet transform to decompose the signal into its separate components.

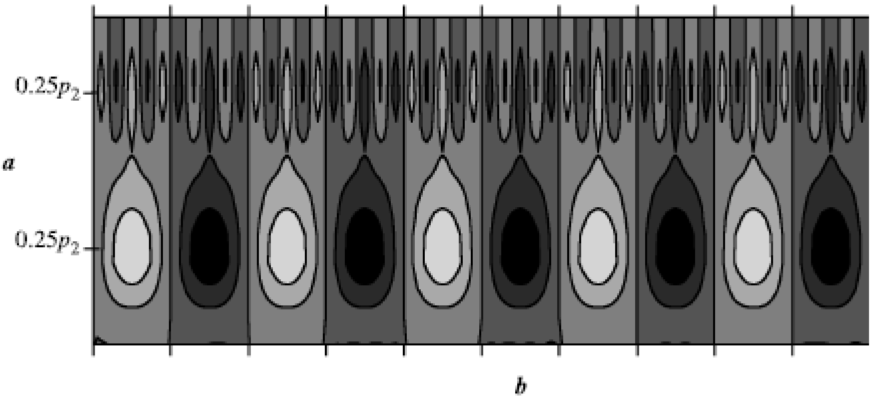

Figure 2.16 Contour plot of T(a,b) for the signal in Figure 2.15.

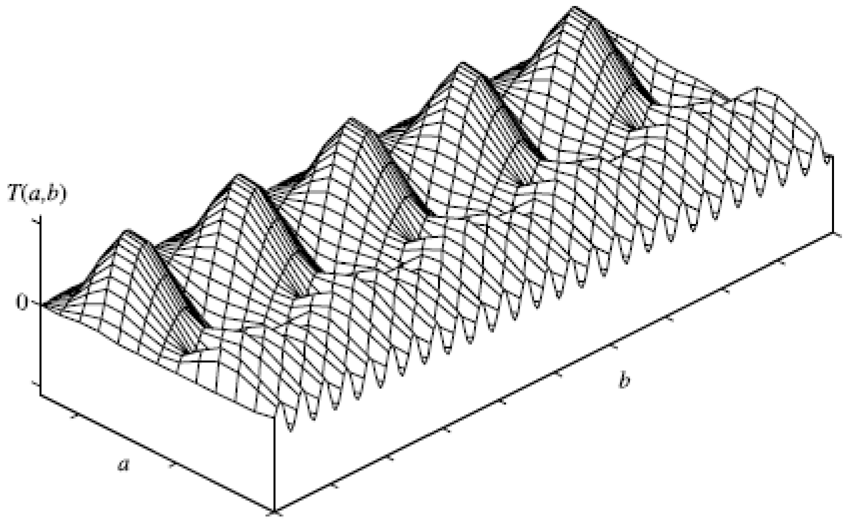

Figure 2.17 Isometric surface plot of T(a,b) for the signal in Figure 2.15.

2.1.1.5 The Inverse Wavelet Transform

The inverse wavelet transform is defined as equation (2.10).

$$x(t) = \frac{1}{C_g}\int_{-\infty}^{\infty}\int_{0}^{\infty} T(a,b)\,\varphi_{a,b}(t)\,\frac{da\,db}{a^2} \quad (2.10)$$

The inverse wavelet transform gives the chance of recover original signal from its wavelet transform. Don’t forget, in inverse wavelet transform, the original wavelet function which is used also in forward transformation is used. If we limit the integration over a range o a values rather than all a values, we can perform a filtering of the original signal.

A very good example of inverse wavelet transform is showed in the figures below. Figure 2.18 is a sinusoidal waveform, Figure 2.19 is an another sinusoidal waveform with a period one quarter of that in Figure 2.18. Figure 2.20 is a burst of high frequency noise and Figure 2.21 is the composed signal obtaining by signals in Figure 2.18, Figure 2.19 and Figure 2.20.

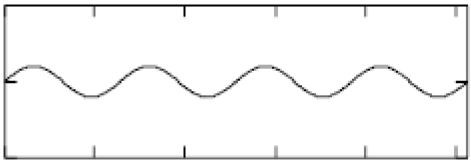

Figure 2.18 Sinusoidal waveform.

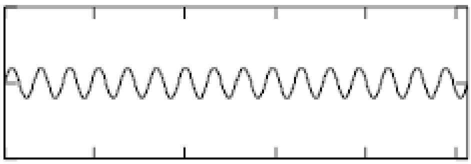

Figure 2.19 Sinusoidal waveform with a period one quarter of that in Figure 2.18.

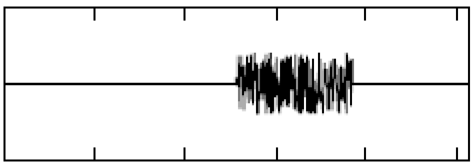

Figure 2.20 High frequency noise.

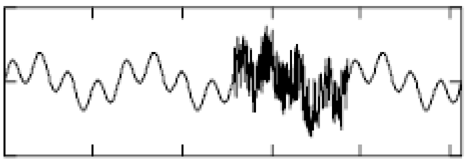

Figure 2.21 Composite signal.

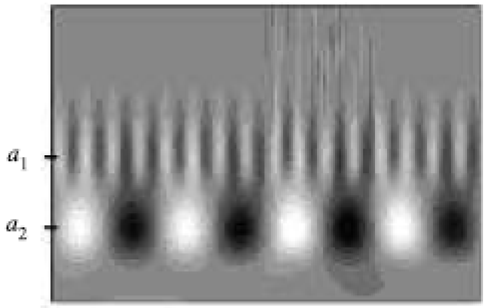

Figure 2.22 Wavelet transform plot of composite signal in Figure 2.21.

Now, if we remove the upper part of transform plot where contains the high frequency component, then we obtain the new plot showed in Figure 2.23. The removed part is limited with the white line.

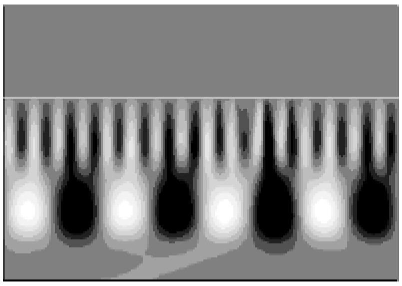

Figure 2.23 Components above the white line removed from transform plot.

If the inverse wavelet transform is applied to plot in Figure 2.23, then we obtain a signal which is the combination of sinusoidal waveforms in Figures 2.18 and Figure 2.19. This obtained composed signal is showed in Figure 2.24.

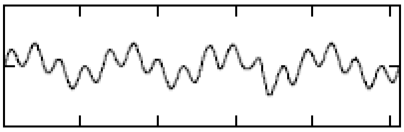

Figure 2.24 Reconstructed signal obtained by the inverse wavelet transform.

2.1.2 Discrete Wavelet Transform

It is computationally impossible to perform continuous wavelet transform on a signal. So for analyzing, discrete wavelet transform and its inverse is used in practice. Performing discrete wavelet transform on a discrete input signals of finite length is more applicable and faster.

We know that wavelet function is defined at scale a and location b as equation (2.11) from the previous section.

$$\varphi_{a,b}(t) = \frac{1}{\sqrt{a}}\varphi\left(\frac{t-b}{a}\right) \tag{2.11}$$

In equation (2.11), a continuous signal x(t) is considered where dilation and location parameters are discrete values. To link a to b, we move in discrete steps of b which are proportional to scale of a. Then new discretization wavelet has the form in equation (2.12) where the integers m and n control dilation and translation respectively.

$$\varphi_{m,n}(t) = \frac{1}{\sqrt{a_0^m}}\varphi\left(\frac{t-nb_0a_0^m}{a_0^m}\right) \tag{2.12}$$

a_0 is fixed dilation parameter which is greater than 1, b_0 is location parameter which is greater than 0. So from the equation (2.12), the size of translation step is $\Delta b = b_0a_0^m$ which is directly proportional to wavelet scale a_0^m. The wavelet transform of continuous signal x(t) by using discrete wavelets from equation (2.12) is showed in equation (2.13).

$$T_{m,n} = \int_{-\infty}^{\infty} x(t)\frac{1}{a_0^{\frac{m}{2}}}\varphi(a_0^{-m}t - nb_0)dt \tag{2.13}$$

Which is also expressed basically as equation (2.14)

$$T_{m,n} = \langle x, \varphi_{m,n}\rangle \tag{2.14}$$

$T_{m,n}$ are discrete wavelet transform values giving on a scale-location grid of index *m,n*. For the discrete wavelet transform, the values $T_{m,n}$ are known as wavelet coefficients or detail coefficients.

CHAPTER THREE
ANALYSIS METHOD

3.1 Artificial Neural Network

Neural network is a kind of system that imitates human brain. A neural network creates connections among input elements and makes an organization between them. This organization and weight of connections determine the output.

Generally, neural networks attempt to reach a specific target output by using a particular input such a situation is shown in Figure 3.1. The network is based on a comparison of the output and the target, until the output matches the target. Many input/target pairs are needed to train a network and obtain the desired result.

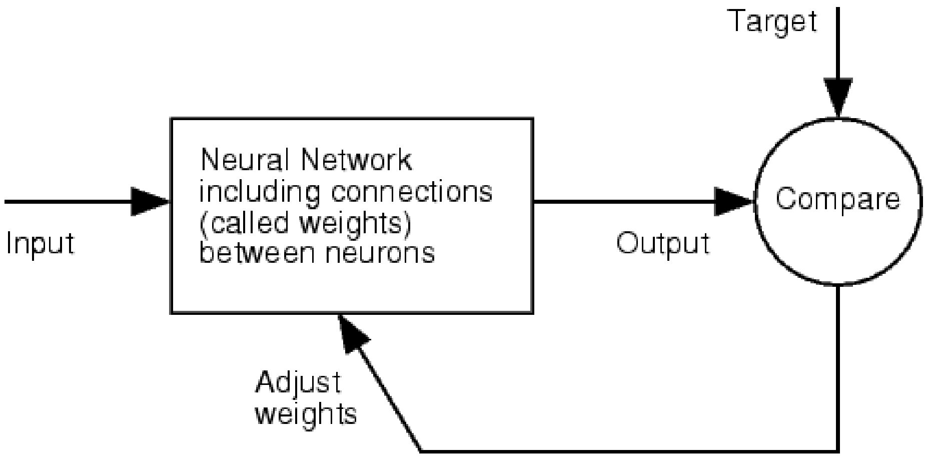

Figure 3.1 Neural Network System.

Neural Networks are currently used in various fields, including voice and image recognition systems, industrial robotics, data mining, engineering, financial, and other practical applications.

3.1.1 Neuron

The fundamental structure for neural networks is the single-input neuron, such as showed in Figure 3.2.

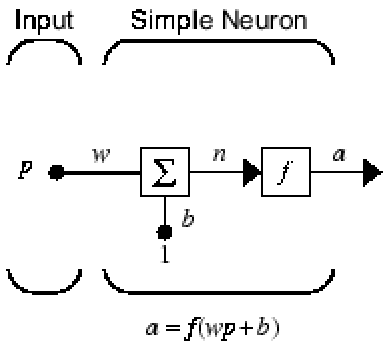

Figure 3.2 A simple neuron.

If we analysis the operation of the neuron structure, we see that firstly scalar input *p* is multiplied by the scalar weight *w*, so *wp* again a scalar. Secondly, the weighted input *wp* is added to the scalar bias *b*. So right now, the net input becomes *(wp+b)* and it is showed as *n*. Finally, the net input is passed through the transfer function *f*, which produces the scalar output *a*. These three processes are named as the weight function, the net input function and the transfer function respectively.

w and *b* are both adjustable scalar parameters of the neuron and these parameters are used to reach desired result. So we can train the network by adjusting the weight or bias parameters.

3.1.2 Transfer Function

There are many transfer functions used in neural network systems but two of them are more common. These are linear transfer function and logsig transfer function.

3.1.2.1 Linear Transfer Function

The Figure 3.3 illustrates the linear transfer function. Neurons of this type are used in the final layer of multilayer networks that are used as function approximators.

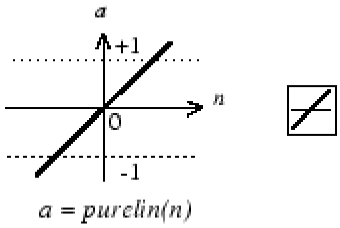

Figure 3.3 Linear transfer function.

3.1.2.2 Log - Sig Transfer Function

The Figure 3.4 illustrates the logsig transfer function which generates outputs between 0 and 1, as takes the input as any value between plus and minus infinity. The logsig transfer function is commonly used in the hidden layers of multilayer networks.

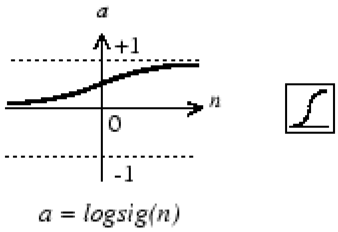

Figure 3.4 Log - Sig moid transfer function.

Note that the symbol in the square on the right side of each transfer function graph represents the associated transfer function. These icons replace the general *f* in the network diagram blocks to show the particular transfer function being used.

3.1.3 Neuron with Vector Input

The simple neuron can be extended to handle inputs that are vectors. A neuron with a single *R*-element input vector is shown in Figure 3.5. Here the individual input elements $P_1, P_2, \ldots P_R$ are multiplied by weights $W_{1,1}, W_{1,2}, \ldots W_{1,R}$ and the weighted values are fed to the summing junction. Their sum is simply *Wp*, the dot product of the single row matrix *W* and the vector *p*.

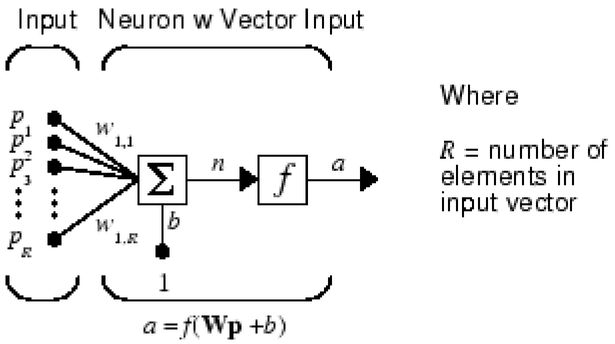

Figure 3.5 Neuron with vector input.

The neuron has a bias *b*, which is summed with the weighted inputs to form the net input *n*. The net input *n* is the argument of the transfer function *f*.

Neuron shown in Figure 3.5 contains a lot of detail. When you consider networks with many neurons, there is so much detail that the main thoughts tend to be lost. Thus, the authors have devised an abbreviated notation for an individual neuron. This notation, which is used later in circuits of multiple neurons, is shown in Figure 3.6.

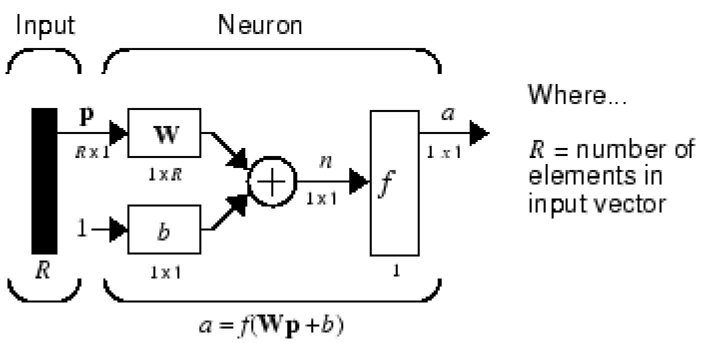

Figure 3.6 Abbreviated notation for an individual neuron.

Here the input vector *p* is represented by the solid dark vertical bar at the left. The dimensions of *p* are shown below the symbol *p* in the figure as $R \times 1$. Thus, *p* is a vector of *R* input elements. These inputs postmultiply the single-row, R-column matrix *W*. As before, a constant 1 enters the neuron as an input and is multiplied by a scalar bias *b*. The net input to the transfer function *f* is *n*, the sum of the bias *b* and the product *Wp*. This sum is passed to the transfer function *f* to get the neuron's output a, which in this case is a scalar. Note that if there were more than one neuron, the network output would be a vector.

Similar to simple neuron, neuron in Figure 3.6 contains three parts, weight function, the net input function and the transfer function. Note that if a specific transfer function is used, the symbol for transfer function replaces by used specific function figure. Some of specific transfer function figures are shown in Figure 3.7.

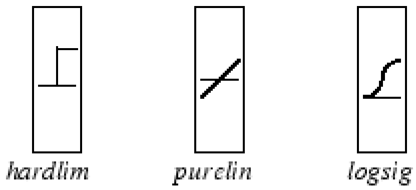

Figure 3.7 Sample function Figures.

Two or more of the neurons can be combined in one layer, and a particular network could contain one or more such layers. Layers are divided to input layer, hidden layer and output layer types. A simple combination is showed in Figure 3.8.

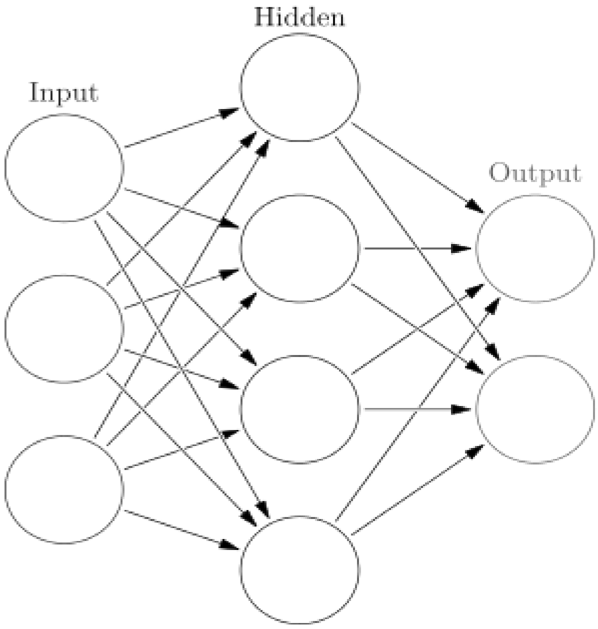

Figure 3.8 Input layer includes three neurons, hidden layer layer includes four neurons and output layer layer includes two neurons.

3.1.4 Multiple Layers of Neurons

A network can have several layers. Each layer has a weight matrix *W*, a bias vector *b*, and an output vector *a*. To distinguish between the weight matrices, output vectors and other content for each of these layers, the number of the layer is appended as a superscript to the variable of interest.

All the layers of a multilayer network have different roles in the system. A layer that produces the network output is called an output layer. All other layers are called hidden layers. For example, for a three-layer network, the third layer is output layer, the first and second layers are hidden layers.

Multiple-layer networks can be more useful and powerful. For example, in a network of two layers, the first and second layers can have different types of functions. Lots of alternatives can be trained in layers to obtain desired result.

3.1.5 Workflow for Neural Network Design

The work flow for the neural network design process has seven primary steps as shown below.

1. Collect data
2. Create the network
3. Configure the network
4. Initialize the weights and biases
5. Train the network
6. Validate the network
7. Use the network

After creating a neural network, it needs to be configured and then trained. We can configure the network by sample data, related to problem that we want to solve. After the network has been configured, we can adjust weight and bias parameters to optimize network performance. This adjustment is referred to as training the network. Configuration and training require that the network be provided with example data.

3.1.6 Application Areas for Neural Network Design

Neural Networks are currently used in various fields, some of good examples are listed in Table 3.1.

Table 3.1 Some example application areas of Neural Network Systems.

Application Area	Usage Explanation
Aerospace	High-performance aircraft autopilot, flight path simulation, aircraft control systems, autopilot enhancements, aircraft component simulation, aircraft component fault detection.
Automotive	Automobile automatic guidance system, warranty activity analysis.
Banking	Check and other document reading, credit application evaluation.
Defense	Weapon steering, target tracking, object discrimination, facial recognition, new kinds of sensors, sonar, radar.
Electronics	Code sequence prediction, integrated circuit chip layout, process control, machine vision, voice synthesis, nonlinear modeling.
Entertainment	Animation, special effects, market forecasting.
Financial	Real estate appraisal, loan advising, mortgage screening, corporate bond rating, credit-line use analysis, portfolio trading program.
Industrial	Neural networks are being trained to predict the output gases of furnaces and other industrial processes. They then replace complex and costly equipment used for this purpose in the past.
Medical	Breast cancer cell analysis, EEG and ECG analysis, prosthesis design, optimization of transplant times.
Robotics	Trajectory control, forklift robot, manipulator controllers, vision systems.
Speech	Speech recognition, speech compression, vowel classification, text-to-speech synthesis.
Telecommunications	Image and data compression, automated information services, real-time translation of spoken language.

CHAPTER FOUR
VOICE RECOGNITION

4.1 Overview

The system will try to recognize voices by help of significant features of the recorded audio signal obtained by discrete time wavelet transform. Discrete wavelet transform based features are coefficients and also energies of them. Feature matrices will be created for every voice sample and they will be analyzed by ‘Artificial Neural Network’ and recognition performance will be determined. Matlab R2012a (7.14.0.739 64-bit version) will be used to improve the program of the thesis.

4.2 Software

The software involves different sections which are aimed to different purposes. These sections are voice recording, wavelet decomposition, data management, neural network training structure, result and performance parts. All these sections have their own software codes and they work in accordance. Main parts of software are listed below;

- Voice recording and saving part.
- Multi-level 1-D wavelet decomposition part.
- Data management part.
- Creating Neural Network part.
- Result and analysis part.

4.2.1 Flowchart of Software

Flowchart in Figure 4.1 shows the main operating logic briefly.

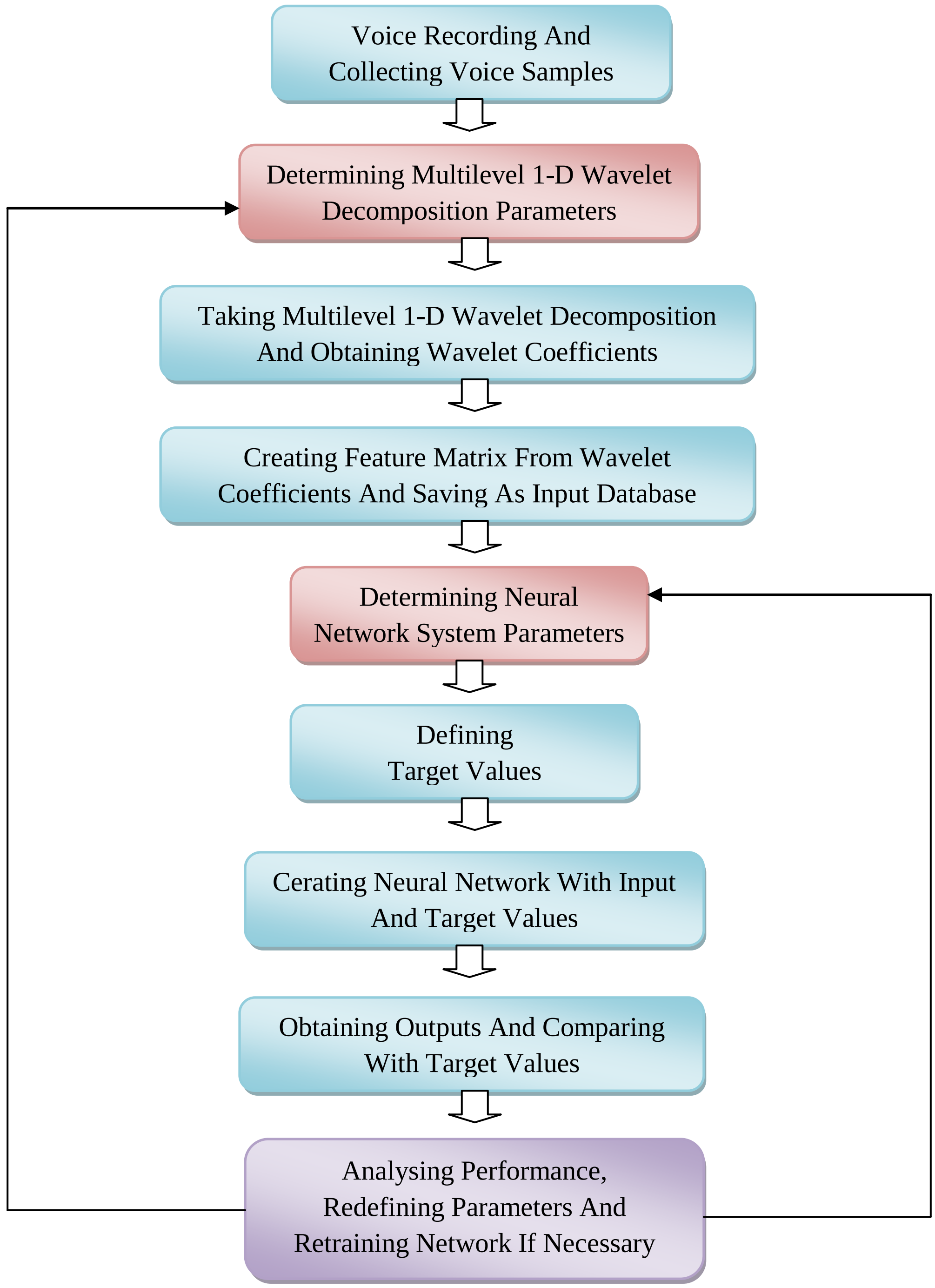

Figure 4.1 Flowchart of the software.

4.2.2 Development of Software

4.2.2.1 Voice Recording

Sampling frequency is selected as 44100 Hz for high quality voice recording, then a very basic Matlab command, ‘wavrecord’ is used to record the voice. for a healthy analysis, all of the recorded voice samples must be in the same format so duration of the samples are adjusted to same time and their possible maximum-minimum values of amplitudes are determined by normalization function. The words are usually take shorter than three seconds so two seconds is enough for each voice sample. Related to this, recording time is adjusted to three seconds, then software determines word and its time interval and takes this part as new sound file in suitable and standard format. Related part of software in follow.

Firstly, determining the sampling frequency, recording voice and normalizing it are showed below.

```
Fs=44100;                          %sampling frequency
voice = wavrecord(3*Fs,Fs);  %voice recording
y=voice/max(abs(voice));       %normalizing voice recorded
```

Following part determines word has been said within recording.

```
%%%Taking the voice part%%%
check1=0;
check2=0;
n=Fs*3;
for j=1:n,
   if ((abs(y(j))>=0.04) && (check1==0))
      begin=j;
      check1=1;
   end
```

```
    if ((abs(y(n+1-j))>=0.01) && (check2==0))
        finish=n-j;
        check2=1;
    end
end
ynew=y(begin:finish);
%%%%%%
```

Then recorded voice sample is adjusted a standard format with a duration of two seconds by following codes.

```
le=1.5/3*length(voice)-length(ynew); %finding length for zeros
len=zeros(le,1);            %creating zeros to complate sound sample to 2 seconds
ynew2=[ynew;len];     %sound sample in 2 seconds length is ready
```

4.2.2.2 Creating Feature Matrix

After recording enough voice samples, all of the voice files are placed in a specific folder named as voices. The feature matrix will be include wavelets coefficient values obtained by multilevel one dimensional (or 1-D) wavelet decomposition method. The feature matrix will be used as main database for the system.

The feature matrix contains wavelet coefficients for every voice file, which are obtained by multilevel 1-D wavelet decomposition (a discrete wavelet transform method). In discrete wavelet transform, suitable wavelets must be used. These are Haar, Daubechies, Symlets, Coiflets, Biorthogonal, Reverse Biorthogonal and Discrete Meyer wavelets. All these wavelets will be used respectively to see their influence in experimental results.

To create feature matrix, the software firstly goes to voice folder and reads all of the voice samples in it. Then multilevel 1-D wavelet decomposition is applied to all of these samples so wavelet coefficients are obtained sample by sample. Multilevel 1-D wavelet analysis generates approximation coefficients by correlating signal with low-pass filter, and generates detail coefficients by correlating signal with high-pass filter. The level of decomposition determines the coefficient structure. For example, for a signal ‘s’, third level multilevel 1-D wavelet decomposition produces one level of approximation coefficients named as cA_3, and the levels of detail coefficients named as cD_1, cD_2 and cD_3. The structure is showed in the following figure.

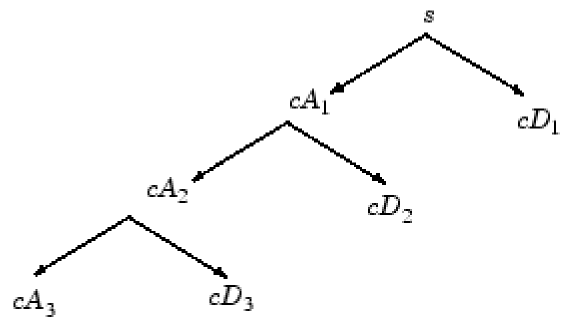

Figure 4.2 A third level 1-D wavelet decomposition structure, shows the terminal nodes of the tree.

A wavelet coefficient is not just a value, it's a collection of values. It tells us that lots of coefficients means huge numbers of values and it's very hard to process them. So it's better to use energy values for each level of coefficients.

Finally, a total feature matrix will be generated by collecting of energy values of determined level 1-D wavelet decomposition coefficients for all voice sample in the folder. Feature matrix is saved as external file and it will be used as input data set for neural network system. In Matlab, *Wavedec* command performs multilevel 1-D wavelet decomposition and *Wenergy* command gives energy values of approximation and detail coefficients of a multilevel 1-D wavelet decomposition. It's more useful and computable to use energies of coefficients because they include less and summary data. Use of these commands and related part of software is explained as follows.

[C,L] = wavedec(X,N,'wname') returns the wavelet decomposition of the signal X at level N, using 'wname'. N must be a strictly positive integer. The output decomposition structure contains the wavelet decomposition vector C and the bookkeeping vector L. The structure is organized as in this level-3 decomposition example.

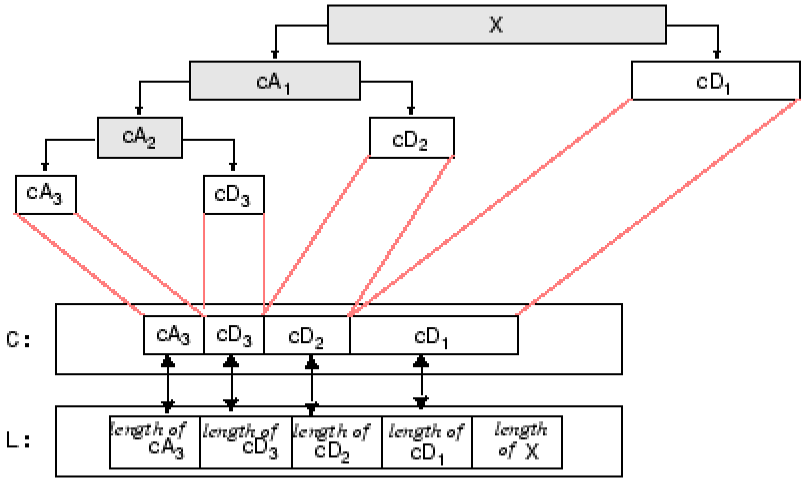

Figure 4.3 Multilevel 1-D wavelet decomposition structure. *X* is input signal, *cA*'s are approximation coefficients (from the low-pass filter), *cD*'s are detail coefficients (from the high-pass filter). *C* and *L* are matrices that include coefficients and length of them respectively.

For a one dimensional wavelet decomposition [C,L], [Ea,Ed] = wenergy(C,L) returns Ea, which is the percentage of energy corresponding to the approximation and Ed, which is the vector containing the percentages of energy corresponding to the details.

Software firstly goes to voice samples folder and read all voice files in it with *wavread* command and takes multi-level one dimensional wavelet decomposition with *wavedec* command. Then find energies of wavelet coefficients with *wenergy* command. After finishing this procedure for all files, it saves obtained energy values as vectors in a file voice by voice with *save* command. Related part of code follows.

```
%%%Reaching directory of wav files and making preparation%%%
cd ('voice')        %directory of samples
delete input.mat     %deleting previous camf.mat
input = [];      %creating empty feature matrix
firstdir=dir;
last=length(firstdir)
lastnew=last-2      %determinig files number in directory
```

```
%%%Reading all wav files in directory%%%
for i=3:last
    nameoffile=firstdir(i).name;
[y,Fs,bits]=wavread(nameoffile);

%%%Taking Multi-level 1-D wavelet decomposition and getting energies of
coefficients%%%
[c,l] = wavedec(y,level,tw); %multi-level 1-D wavelet decomposition
[Ea,Ed] = wenergy(c,l);      %finding energies of coefficients

%%%Creating feature matrix%%%
   input=[input; Ea Ed]
   save (strcat('input','.mat'),'input');
end
```

Saved file includes input data for neural network system. Also target data must be determined related to these input data. Target data is created as a vector with the commands below;

```
%%%Creating target data vector%%%
targetfeat = [];
targetfeat = [1:1:lastnew]
save (strcat('target','.mat'),'targetfeat');
```

4.2.2.3 Design of Neural Network

Matlab has four different main neural network application areas. These are function fitting, pattern recognition, clustering, and time series analysis. The work flow for any of these problems has seven primary steps.

4.2.2.3.1 Fitting Function. Neural networks are good at fitting functions. To define a fitting function problem, arrange a set of input vectors as columns in a matrix and then, arrange another set of target vectors (the correct output vectors for each of the input vectors) into a second matrix and create a relation system between them.

4.2.2.3.2 Recognizing Pattern. In addition to function fitting, neural networks are also good at recognizing patterns. To define a pattern recognition problem, arrange a set of input vectors as columns in a matrix, then arrange another set of target vectors so that they indicate the classes to which the input vectors are assigned. There are two approaches to creating the target vectors.

One approach can be used when there are only two classes; you set each scalar target value to either 1 or 0, indicating which class the corresponding input belongs to. Second approach, target vectors can have N elements, where for each target vector, one element is 1 and the others are 0. This defines a problem where inputs are to be classified into N different classes.

4.2.2.3.3 Clustering Data. Clustering data is another excellent application for neural networks. This process involves grouping data by similarity. To define a clustering problem, simply arrange input vectors to be clustered as columns in an input matrix.

4.2.2.3.4 Time Series Prediction. Dynamic neural networks are good at time series prediction. To define a time series problem for the toolbox, arrange a set of input vectors as columns in a cell array. Then, arrange another set of target vectors (the correct output vectors for each of the input vectors) into a second cell array However, there are cases in which you only need to have a target data set.

In this thesis, the most suitable neural network type is fitting function because software bases on input and target matrices and creating a relation system between them.

In a basic fitting function based neural network system, firstly input and target values are determined. Then percentages of training, validation and test data are set. Finally the number of hidden neurons are defined and the network is trained to fit input and targets. Related software code is in below;

```
%%%Neural Network System%%%
validation=(100-training)/2; %determining validation percentage
testing=validation;          %determining testing percentage

cd ('voice')
load input.mat      %loading input data
cd ..
load target.mat     %loading target data
inputs = input';
targets = targetfeat;

% Create a Fitting Network
hiddenLayerSize = hidden; %defining hidden layer size
net = fitnet(hiddenLayerSize);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = training/100; %setting training data ratio
net.divideParam.valRatio = validation/100; %setting validation data ratio
net.divideParam.testRatio = testing/100; %setting testing data ratio
% Train the Network
[net,tr] = train(net,inputs,targets);

% Test the Network
outputs = net(inputs); %obtaining output data
errors = gsubtract(targets,outputs); %errors of network
performance = perform(net,targets,outputs); %performance of network
result=round(outputs'); %result data obtained from output data
```

```
% Testing accuracy of Network and Retraining if network is not efficient.
kar=[targets',result];
counter=0;
for i = 1:lastnew
    if (kar(i,1)-kar(i,2))==0
        counter=counter+1;
    else counter=counter;
    end
end
accuracy=counter/lastnew;
if accuracy<0.30
    run nn
else
save (strcat('outputs','.mat'),'outputs');

% Showing summary results for a efficiency is enough.
set(handles.edit2 , 'string' , accuracy);
set(handles.edit3 , 'string' , counter);
set(handles.edit4 , 'string' , lastnew);
end
```

4.2.2.4 Development of User Interface

Matlab has a very useful tool to design user interfaces, which is named as Guide. By the help of this tool, the system becomes manageable. Push buttons, popup menus, text boxes, sliders, panels…etc helps user to manage system without writing any code. Also interface can be design visually for easy usage.

Software codes that explained in previous sections are a part of the main software and main software has a visual interface. This interface is designed by Matlab Guide. View of visual interface is shown in Figure 4.4.

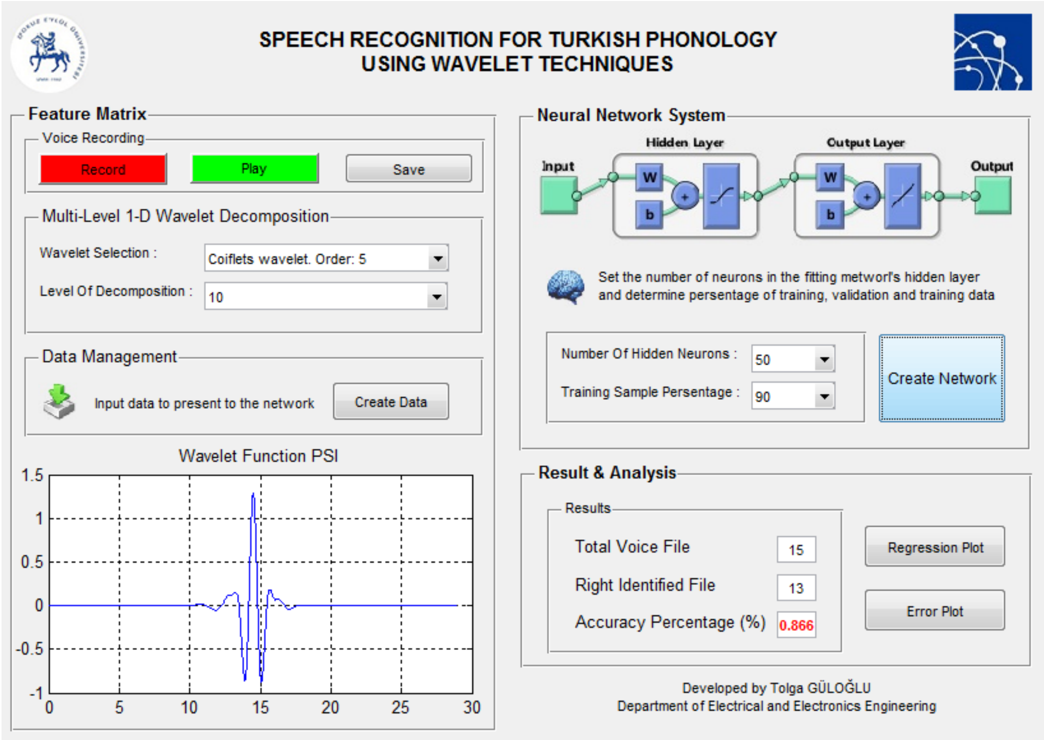

Figure 4.4 View of user interface.

User interface has three main sections. These are feature matrix, neural network system and result & analysis. Feature matrix section has three different panels in itself, these are voice recording panel, multi-level one dimensional wavelet decomposition panel, data management panel and selected wavelet show panel. Let's see functions of these section.

4.2.2.4.1 Voice Recording. In this panel, you can easily record voice, listen it and save it as wav file by the help of the buttons. The panel uses codes that explained in 4.2.2.1 Voice Recording. View of voice recording panel is shown in Figure 4.5.

Figure 4.5 View of voice recording panel.

4.2.2.4.2 Multi-level 1-D Wavelet Decomposition. In this panel, parameters of multi-level one dimensional wavelet decomposition are determined. The first selection is wavelet type, selected wavelet will be used in wavelet decomposition computation. Second selection is level of decomposition which determines the level and structure of approximation and detail coefficients. View of this panel is shown in Figure 4.6.

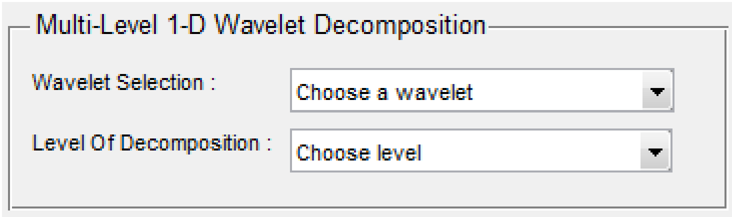

Figure 4.6 View of multi-level 1-d wavelet decomposition panel.

4.2.2.4.3 Data Management. In this panel, obtained values in previous wavelet decomposition panel are converted to a data file which will be used as input data for neural network system. View of data management panel is shown in Figure 4.7.

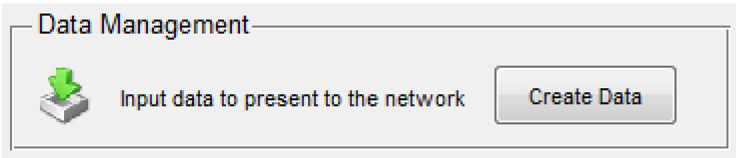

Figure 4.7 View of data management panel.

4.2.2.4.4 Selected Wavelet Monitor. This panel shows us wavelet graph which is selected in multi-level 1-d wavelet decomposition panel. Figure 4.8 shows wavelet monitor when a third order Coiflets wavelet selected.

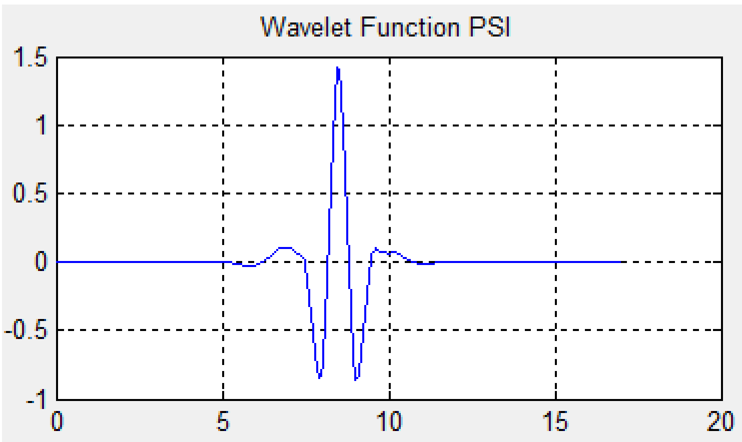

Figure 4.8 View of selected wavelet monitor.

4.2.2.4.5 Neural Network System. This panel gives the opportunity of changing the values of variables of neural network system. These variables are number of hidden neurons and percentage of training, validation and test data. After finishing of all necessary selection, *create network* button trains neural network and creates outputs of the system. Figure 4.9 shows neural network system panel.

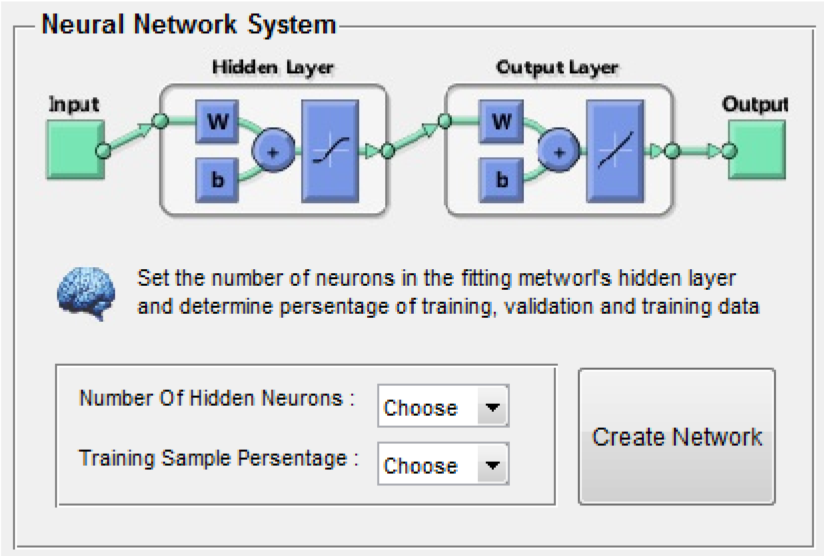

Figure 4.9 View of neural network system panel.

4.2.2.4.6 Result & Analysis. This panel gives statistical information about results of neural network training. Also you can plot error histogram and comparison of output and target values by related buttons. Figure 4.10 shows result and analysis panel.

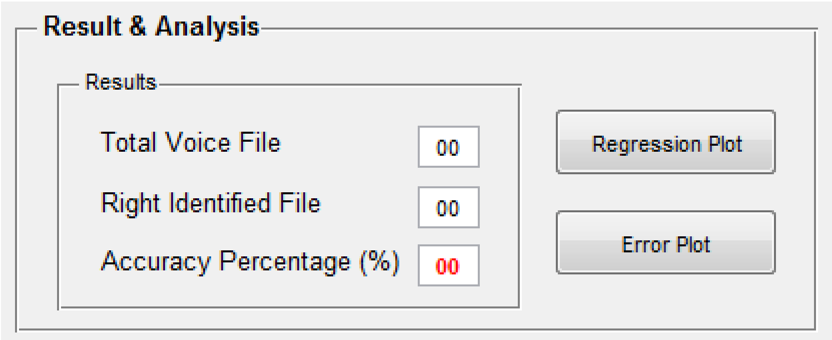

Figure 4.10 View of result and analysis panel.

CHAPTER FIVE
EXPERIMENTAL RESULTS

5.1 Overview

Main idea of thesis is analyzing and comparing voice recognition performance of different wavelets on wavelet transform by using neural network system, The most efficient conditions for wavelet transform and structure of neural network will be uncovered by lost of experiments. Also different types of voice structures will be used to see performance of network. For example monosyllabic, disyllabic, trisyllabic and quadrisyllabic words will be analyzed separately and you will able compare performance of network on these categories.

In experiments, all input words will be associated with a target number value. Wavelet transform based neural network system will try to set up relation between input and target values. Output values and comparing them with target values will give us the performance and accuracy of the system.

5.2 Experiments

Firstly, the most efficient values for following variables will be determined. Then they will be kept fixed and only wavelet type and its order will be changeable to make a detailed analysis on principal variable, wavelet. By the way, different voice structures will be used for all experiments.

- Wavelet decomposition parameters
- Hidden neurons of neural network
- Training, validation and test data percentage of neural network

Related pre-experimental works, it’s decided that the most appropriate level for multi-level one dimensional wavelet decomposition is ten. Because it provides sufficient performance in neural network and it has few values for easy computation in neural network building process.

For neural network training process, we have two main variables. These are number of hidden neurons and the percentage of training, validation and test data.

We know that as number of the hidden neurons increases, performance of neural networks is also increases, but high numbers of hidden neurons means more processing and calculations. Pre-experiments showed that one hundred hidden neurons provides satisfactory results in training neural network and its process time is reasonable so number of hidden neurons is selected as one hundred.

The last variable is the percentage of training, validation and test data used during training of neural network. Pre-experimental results showed that performance of neural network increases as the percentage of training data increases. So we set the percentage of training as ninety percent and we set the percentage of both validation and test data as five percent.

In brief, the following variables have fixed values for all experiments as shown as below;

- Level of one dimensional wavelet decomposition : 10
- Number of hidden neurons of neural network : 100
- Training data percentage of neural network training : 90%
- Validation data percentage of neural network training : 5%
- Test data percentage of neural network training : 5%

The main changeable variable of experiments is the wavelet type. Experiments will be repeated with different types of wavelet in order. Fifteen different recorded words are used in each experiment.

By the way, some of the wavelets have similar or exactly the same structure related their order. For example, the first order of Daubechies wavelet and Biorthogonal wavelet in order 1.1 are exactly the same as can be seen in Figure 5.1 plotted in Matlab with the following codes.

```
%%% Sub-plots of two different wavelets %%%
[phi,psi,xval] = wavefun('db1',10);
subplot(1,2,1)
plot(xval,psi); title('First Order Daubechies Wavelet'); grid on;
[phi1,psi1,phi2,psi2,xval] = wavefun('bior1.1',10)
subplot(1,2,2)
plot(xval,psi1); title('Biorthogonal Wavelet in Order 1.1'); grid on;
```

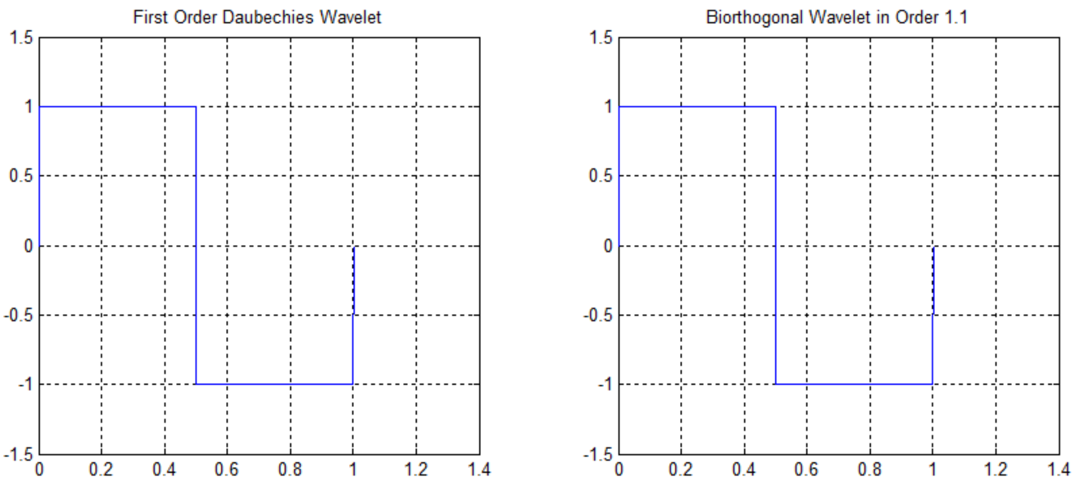

Figure 5.1 Plots of the first order Daubechies wavelet and Biorthogonal wavelet in order 1.1.

So it’s not necessary to use both of these wavelets in experiments. Using only one of them is sufficient.

Also some orders of the some wavelets can be very similar. This similarity causes very close performance in neural network training. For example, ninth and tenth orders of Daubechies wavelets are very similar as can be seen in Figure 5.2 plotted in Matlab with the following codes.

```
%%% Sub-plots of two different wavelets %%%
[phi,psi,xval] = wavefun('db9',10);
subplot(1,2,1)
plot(xval,psi); title('Ninth Order Daubechies Wavelet'); grid on;
[phi,psi,xval] = wavefun('db10',10)
subplot(1,2,2)
plot(xval,psi); title('Tenth Order Daubechies Wavelet'); grid on;
```

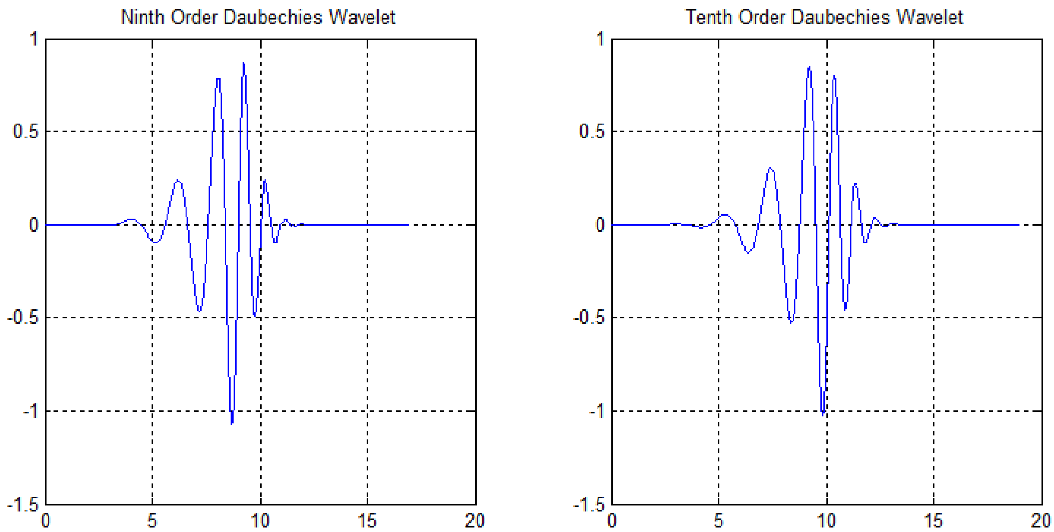

Figure 5.2 Plots of the ninth and tenth orders of Daubechies wavelet.

Because of this similarity, it's not necessary to use both of these wavelets in experiments. So dissimilar wavelets are chosen and used in experiments for more efficient study.

5.2.1 Monosyllabic Word Based Neural Network Experiments

In this experiment, monosyllabic word voice samples are used as input data. All words include three letters and their letter sequences are similar. Letters of words are consonant, vowel and consonant respectively. Used words are listed in Table 5.1.

Table 5.1 Used monosyllabic word voice samples in order.

Word	Target Value		Word	Target Value		Word	Target Value
bal	1		git	6		pir	11
cin	2		hız	7		sen	12
çığ	3		kim	8		şan	13
dök	4		mil	9		tur	14
fiş	5		net	10		yüz	15

Table 5.2 Experimental results of monosyllabic word based neural network experiments.

Wavelet	Order	Test 1	Test 2	Test 3	Test 4	Test 5	Average	Difference
Coiflets	3	100%	87%	87%	93%	87%	**91%**	13%
Biorthogonal	1.5	93%	87%	87%	93%	87%	**89%**	7%
Biorthogonal	2.4	87%	87%	87%	87%	93%	**88%**	7%
Coiflets	5	87%	87%	87%	87%	93%	**88%**	7%
Coiflets	1	87%	87%	87%	93%	87%	**88%**	7%
Daubechies	10	87%	87%	87%	87%	87%	**87%**	0%
Reverse-Bio	2.4	87%	87%	87%	87%	87%	**87%**	0%
Symlet	8	87%	87%	87%	87%	87%	**87%**	0%
Daubechies	1	80%	87%	87%	87%	87%	**85%**	7%
Symlet	2	87%	87%	73%	87%	87%	**84%**	13%
Daubechies	2	87%	87%	87%	67%	87%	**83%**	20%
Symlet	5	60%	87%	87%	87%	87%	**81%**	27%
Daubechies	6	53%	87%	87%	87%	87%	**80%**	33%
Discrete' Meyer	-	40%	87%	87%	87%	93%	**79%**	53%
Biorthogonal	3.9	87%	87%	40%	87%	87%	**77%**	47%
Reverse-Bio	3.3	87%	40%	87%	87%	60%	**72%**	47%
Reverse-Bio	1.5	100%	87%	47%	87%	40%	**72%**	60%
Overall Average		-	-	-	-	-	**83%**	20%

Words are input data and target numbers are associated values in neural network training. After training neural network, comparison of target and output data will give us performance of the wavelet that used in related test.

After completing the first experiment, 83% average accuracy performance is obtained for used all wavelets. The third order Daubechies wavelet achieved the highest average accuracy performance among used wavelets with value of 91%. The highest accuracy performance of tests is 100% which is obtained by The third order Daubechies and Reverse- Biorthogonal 1.5 wavelets.

The tenth order Daubechies, eighth order Symlet and Reverse-Biorthogonal 2.4 wavelets achieved the most stable performance results. They obtained the same accuracy perforamance in all tests. Performance of Reverse-Biorthogonal 2.4 wavelet was the most unstable. Difference of its performance percentage achieved 60%.

One of the 100% accuracy performance belongs to the third order Daubechies wavelet used test. For related test, correct detections can be seen on diagonal line in regression plot of neural network which is showed in Figure 5.3. There is no incorrect detections.

Figure 5.4 shows the error histogram of the same test. All detections are accumulated around zero error line and have short distances to this line. So all the result values can be considered accurate.

Table 5.3 Comparison of target and result values.

Target Values	**Output Values**	**Deviation**	**Explanation**	**Improved Result Values**
1	1	0	Correct	1
2	2	0	Correct	2
3	3.4269	-0.4269	Negligible	3
4	4	0	Correct	4
5	5	0	Correct	5
6	6	0	Correct	6
7	7.0725	-0.0725	Negligible	7
8	8	0	Correct	8
9	9	0	Correct	9
10	10	0	Correct	10
11	11	0	Correct	11
12	12	0	Correct	12
13	13	0	Correct	13
14	14	0	Correct	14
15	15	0	Correct	15

Related to Table 5.3, it can be easily seen that all target and result values match, so there is no mismatching values for this test.

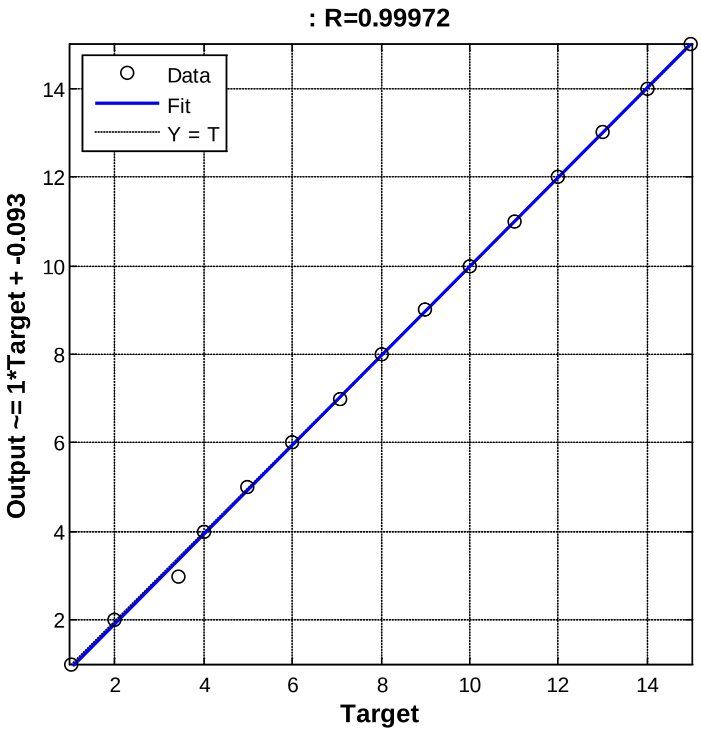

Figure 5.3 Regression plot of neural network.

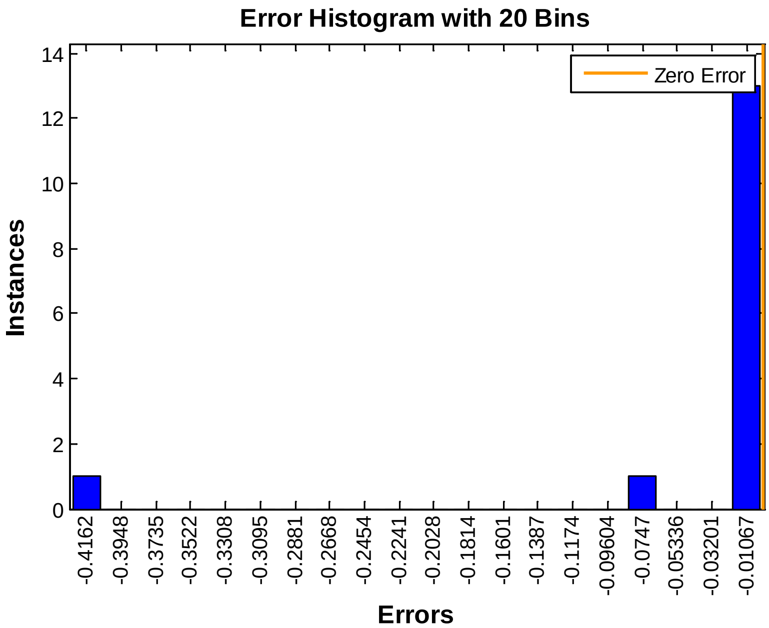

Figure 5.4 Error histogram plot of neural network.

5.2.2 Disyllabic Word Based Neural Network Experiments

In these experiments, disyllabic word voice samples are used as input data. Related to information that explained about Turkish phonology in the first chapter, disyllabic words will be categorized in the following experiments. Because vowels and consonants have different voice characteristic related to their positions in a word.

5.2.2.1 Starting and Ending with Vowel Words Experiments

The first disyllabic word type is starting and ending with vowel words. Used words are listed in Table 5.4.

Table 5.4 Used disyllabic word voice samples in order.

Word	Target Value	Word	Target Value	Word	Target Value
ada	1	efe	6	ölçü	11
algı	2	etki	7	örgü	12
aşçı	3	ilgi	8	uyku	13
atkı	4	imge	9	ülke	14
ece	5	ordu	10	ütü	15

After completing this experiment, the tenth order Daubechies wavelet achieve the best average accuracy performance with 89%. The highest accuracy performance of tests is 93% which is obtained by various wavelets. 93% accuracy performance means fourteen correct and only one incorrect detections in total fifteen samples.

The second order Daubechies wavelet achieved the most stable performance result. It obtained the same accuracy performance in all tests. Performance of Biorthogonal 1.5 wavelet was the most unstable. Difference of its performance percentage achieved 53%.

One of the 93% accuracy performance belongs to the test that tenth order Daubechies wavelet used. For related test, correct detections can be seen on diagonal line in regression plot of neural network which is showed in Figure 5.5. Incorrect detection can be seen out of diagonal line in the same figure.

Table 5.5 Results of neural network experiments for disyllabic words starting and ending with vowels.

Wavelet	Order	Test 1	Test 2	Test 3	Test 4	Test 5	Average	Difference
Daubechies	10	93%	87%	93%	87%	87%	**89%**	7%
Reverse-Bio	2.4	87%	87%	93%	87%	87%	**88%**	7%
Reverse-Bio	3.3	87%	87%	87%	93%	87%	**88%**	7%
Daubechies	2	87%	87%	87%	87%	87%	**87%**	0%
Biorthogonal	2.4	87%	87%	87%	87%	80%	**85%**	7%
Daubechies	6	87%	87%	87%	87%	73%	**84%**	13%
Coiflets	1	87%	87%	87%	93%	67%	**84%**	27%
Symlet	2	87%	87%	67%	87%	87%	**83%**	20%
Daubechies	1	87%	73%	73%	87%	87%	**81%**	13%
Coiflets	5	87%	60%	87%	87%	87%	**81%**	27%
Biorthogonal	1.5	87%	93%	40%	87%	87%	**79%**	53%
Symlet	8	47%	87%	87%	87%	87%	**79%**	40%
Biorthogonal	3.9	87%	87%	40%	87%	87%	**77%**	47%
Coiflets	3	87%	87%	40%	67%	87%	**73%**	47%
Symlet	5	87%	87%	53%	87%	47%	**72%**	40%
Reverse-Bio	1.5	87%	47%	87%	47%	87%	**71%**	40%
Discrete' Meyer	-	47%	67%	73%	87%	67%	**68%**	40%
Overall Average		-	-	-	-	-	**80%**	25%

Table 5.6 Comparison of target and result values.

Target Values	Output Values	Deviation	Explanation	Improved Result Values
1	1	0	Correct	1
2	2	0	Correct	2
3	2.5737	0.4263	Negligible	3
4	4	0	Correct	4
5	5	0	Correct	9
6	6	0	Correct	6
7	7	0	Correct	7
8	8	0	Correct	8
9	**2.5190**	**6.4810**	**Incorrect**	**9**
10	10	0	Correct	10
11	11	0	Correct	11
12	12	0	Correct	12
13	13	0	Correct	13
14	14	0	Correct	14
15	15	0	Correct	15

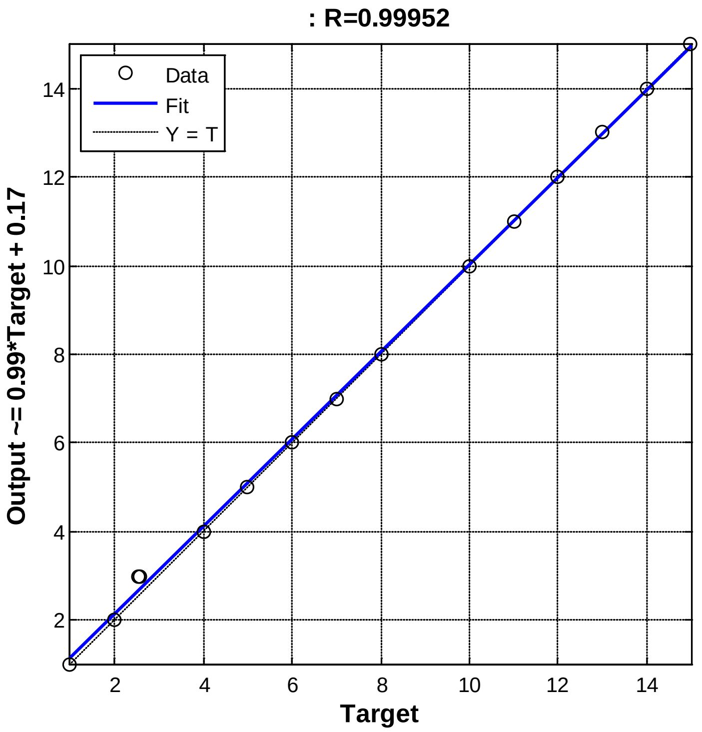

Figure 5.5 Regression plot of neural network.

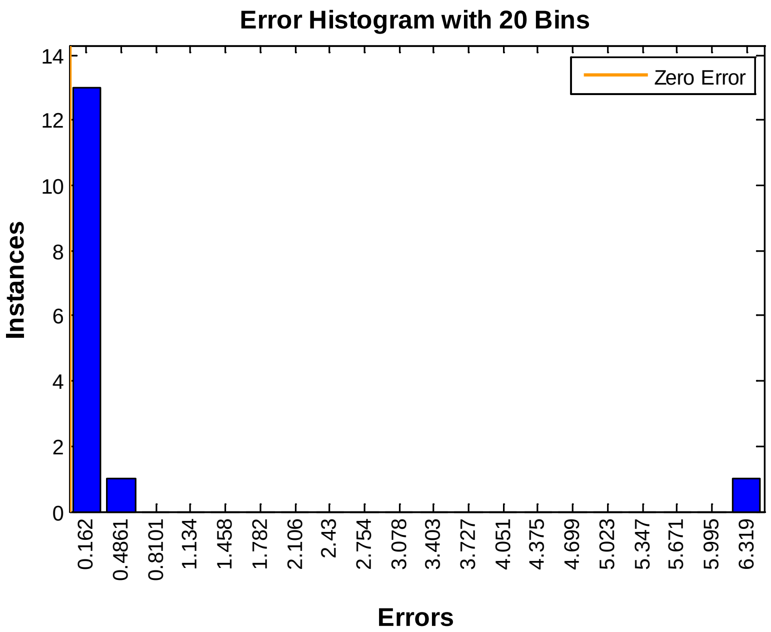

Figure 5.6 Error histogram plot of neural network.

Figure 5.6 shows the error histogram of the same test. Correct detections are accumulated around zero error line and one incorrect detection is accumulated at nearly six units away distance to zero line in the figure.

Mismatch is can be seen in Table 5.6 very easily which is shown in bold writing. Incorrect detection is belong to target value of 9. Equivalent word for this values is ‘imge’ and mismatched with the word ‘aşçı’.

5.2.2.2 Starting with Vowel and Ending Consonant Words Experiments

The second disyllabic word type is starting vowel and ending with consonant words. Used words are listed in Table 5.7 and obtained accuracy performance results are listed in Table 5.8.

Table 5.7 Used disyllabic word voice samples in order.

Word	Target Value	Word	Target Value	Word	Target Value
akıl	1	iklim	6	izmir	11
asker	2	incir	7	okul	12
atak	3	irem	8	olgun	13
ekim	4	ırmak	9	örgün	14
erkek	5	ıslık	10	övünç	15

Related to result Table 5.8, the third order Coiflets wavelet achieve the best average accuracy performance value with 89%. The highest accuracy performance of tests is 93% which is obtained by various wavelets. 93% accuracy performance means fourteen correct and only one incorrect detections in total fifteen samples.

The first order Coiflets wavelet achieved the most stable performance result. It obtained very close accuracy performance in all tests and difference between its highest and lowest value is 7%. Performance of Biorthogonal 1.5, Biorthogonal 3.9 and Reverse- Biorthogonal 2.4 wavelets were the most unstable. Difference of their performance percentage achieved 53%.

Table 5.8 Results of neural network experiments for disyllabic starting with vowel and ending consonant words.

Wavelet	Order	Test 1	Test 2	Test 3	Test 4	Test 5	Average	Difference
Coiflets	3	87%	93%	93%	93%	80%	**89%**	13%
Symlet	8	87%	87%	93%	87%	87%	**88%**	7%
Coiflets	1	87%	87%	80%	87%	87%	**85%**	7%
Coiflets	5	87%	87%	93%	73%	87%	**85%**	20%
Reverse-Bio	3.3	87%	87%	73%	87%	87%	**84%**	13%
Daubechies	2	93%	87%	53%	87%	93%	**83%**	40%
Reverse-Bio	1.5	87%	87%	60%	87%	80%	**80%**	27%
Biorthogonal	2.4	93%	87%	47%	87%	87%	**80%**	47%
Biorthogonal	1.5	87%	87%	87%	40%	93%	**79%**	53%
Daubechies	6	87%	87%	47%	87%	87%	**79%**	40%
Symlet	2	87%	87%	40%	87%	87%	**77%**	47%
Daubechies	10	87%	87%	63%	93%	53%	**77%**	40%
Reverse-Bio	2.4	87%	87%	33%	87%	87%	**76%**	53%
Daubechies	1	87%	47%	53%	87%	87%	**72%**	40%
Discrete' Meyer	-	53%	87%	87%	47%	87%	**72%**	40%
Biorthogonal	3.9	40%	87%	50%	87%	93%	**71%**	53%
Symlet	5	87%	53%	53%	40%	87%	**64%**	47%
Overall Average		-	-	-	-	-	**79%**	34%

One of the 93% accuracy performance belongs to the test that third order Coiflets wavelet used. For related test, correct detections can be seen on diagonal line in regression plot of neural network which is showed in Figure 5.7. Incorrect detection can be seen out of diagonal line in the same figure.

Figure 5.8 shows the error histogram of the same test. Correct detections are accumulated around zero error line and one incorrect detection is accumulated at nearly three units away distance to zero line in the figure.

Mismatch is can be seen in Table 5.9 very easily which is shown in bold writing. Incorrect detection is belong to target value of 10. Equivalent word for this values is ‘ıslık’ and mismatched with the word ‘incir’.

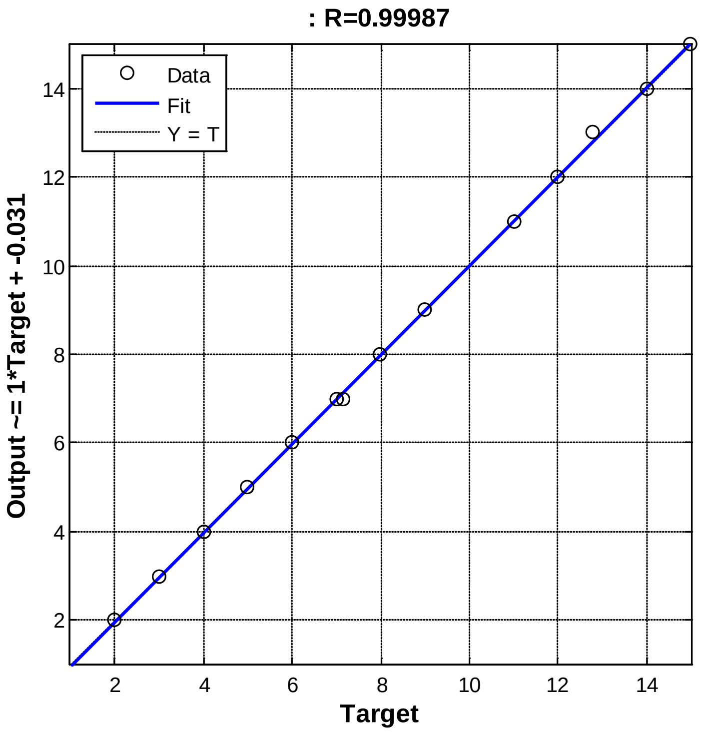

Figure 5.7 Regression plot of neural network.

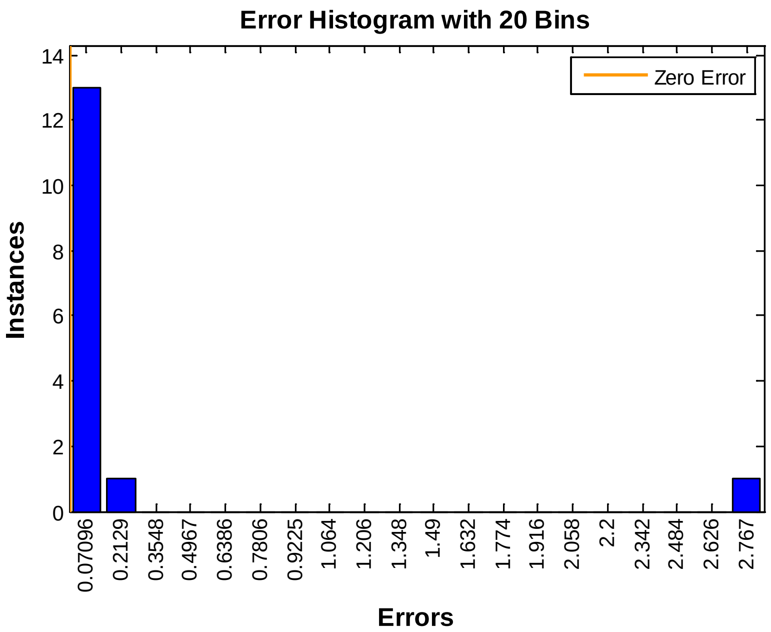

Figure 5.8 Error histogram plot of neural network.

Table 5.9 Comparison of target and result values.

Target Values	Output Values	Deviation	Explanation	Improved Result Values
1	1	0	Correct	1
2	2	0	Correct	2
3	3	0	Correct	3
4	4	0	Correct	4
5	5	0	Correct	5
6	6	0	Correct	6
7	7	0	Correct	7
8	8	0	Correct	8
9	9	0	Correct	9
10	**7.1616**	**2.8384**	**Incorrect**	**7**
11	11	0	Correct	11
12	12	0	Correct	12
13	12.7746	0.2254	Negligible	13
14	14	0	Correct	14
15	15	0	Correct	15

5.2.2.3 Starting with Consonant and Ending Vowel Words Experiments

The third disyllabic word type is starting with consonant and ending with vowel words. Used words are listed in Table 5.10 and obtained accuracy performance results are listed in Table 5.11.

Table 5.10 Used disyllabic word voice samples in order.

Word	Target Value		Word	Target Value		Word	Target Value
balta	1		kapı	6		sezgi	11
cami	2		kıyı	7		şaka	12
çete	3		leke	8		tilki	13
dergi	4		nine	9		tolga	14
gemi	5		pala	10		vazo	15

Related to result Table 5.11, the fifth order Symlet wavelet achieve the best average accuracy performance value with 89%. The highest accuracy performance of tests is 93% which is obtained by various wavelets. 93% accuracy performance means fourteen correct and only one incorrect detections in total fifteen samples.

The third order Coiflets wavelet achieved the most stable performance result with the same accuracy performance in all tests. Performance of second and eighth order Symlet wavelets were the most unstable. Difference of their performance percentage achieved 60%.

Table 5.11 Results of neural network experiments for disyllabic starting with consonant and ending vowel words.

Wavelet	Order	Test 1	Test 2	Test 3	Test 4	Test 5	Average	Difference
Symlet	5	87%	87%	93%	93%	87%	**89%**	7%
Coiflets	3	87%	87%	87%	87%	87%	**87%**	0%
Reverse-Bio	1.5	87%	87%	87%	87%	60%	**81%**	27%
Discrete' Meyer	-	87%	87%	87%	87%	53%	**80%**	33%
Daubechies	10	87%	87%	87%	40%	93%	**79%**	53%
Coiflets	5	87%	40%	87%	87%	87%	**77%**	47%
Daubechies	6	87%	87%	53%	87%	60%	**75%**	33%
Daubechies	1	87%	40%	73%	87%	87%	**75%**	47%
Coiflets	1	87%	87%	47%	67%	87%	**75%**	40%
Biorthogonal	3.9	87%	87%	40%	67%	87%	**73%**	47%
Reverse-Bio	3.3	33%	73%	87%	87%	87%	**73%**	53%
Biorthogonal	1.5	80%	47%	60%	87%	87%	**72%**	40%
Symlet	2	93%	87%	87%	60%	33%	**72%**	60%
Daubechies	2	87%	60%	87%	73%	47%	**71%**	40%
Reverse-Bio	2.4	47%	87%	73%	53%	87%	**69%**	40%
Symlet	8	93%	87%	47%	33%	47%	**61%**	60%
Biorthogonal	2.4	33%	47%	87%	87%	40%	**59%**	53%
Overall Average		-	-	-	-	-	**75%**	40%

One of the 93% accuracy performance belongs to the test fifth order Symlet wavelet used. For related test, correct detections can be seen on diagonal line in regression plot of neural network which is showed in Figure 5.9. Incorrect detection can be seen out of diagonal line in the same figure.

Figure 5.10 shows the error histogram of the same test. Correct detections are accumulated around zero error line and one incorrect detection is accumulated at nearly five units away distance to zero line in the figure.

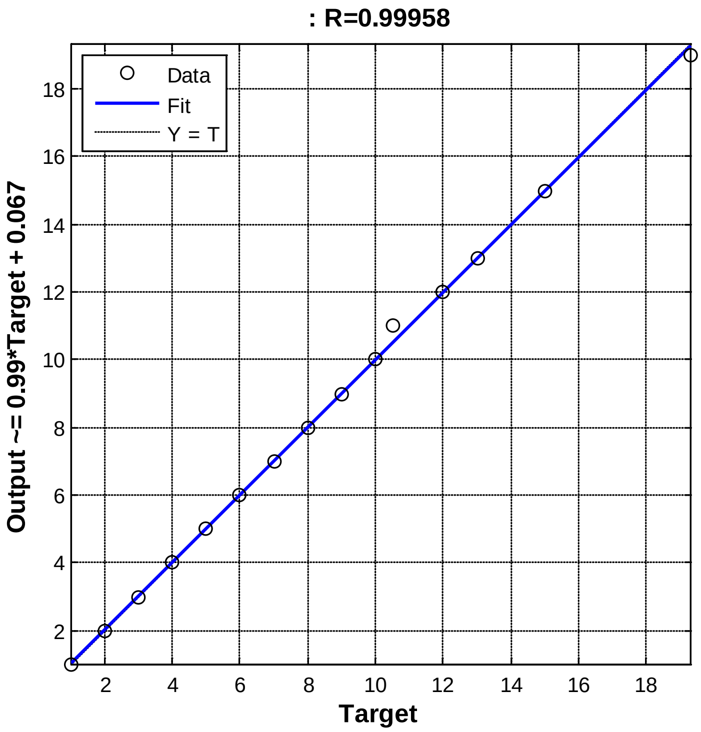

Figure 5.9 Regression plot of neural network.

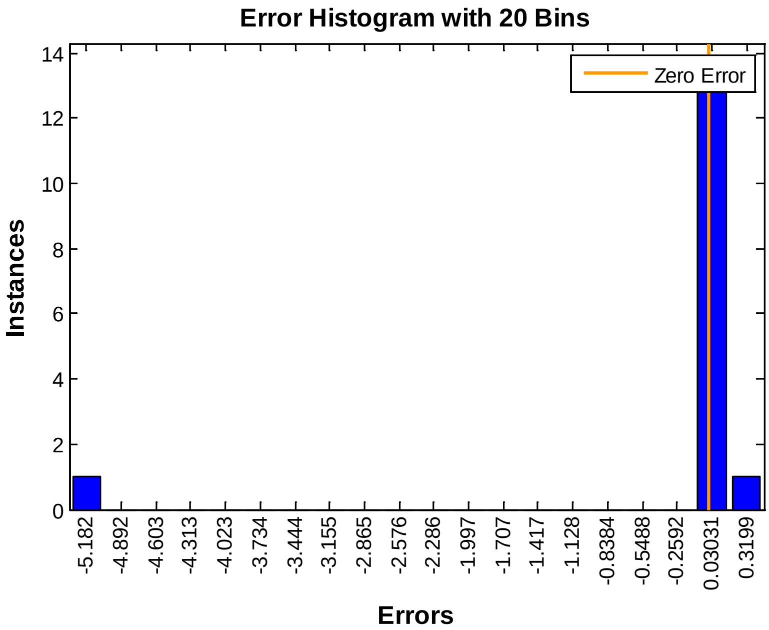

Figure 5.10 Error histogram plot of neural network.

Table 5.12 Comparison of target and result values.

Target Values	Output Values	Deviation	Explanation	Improved Result Values
1	1	0	Correct	1
2	2	0	Correct	2
3	3	0	Correct	3
4	4	0	Correct	4
5	5	0	Correct	5
6	6	0	Correct	6
7	7	0	Correct	7
8	8	0	Correct	8
9	9	0	Correct	9
10	10	0	Correct	10
11	10.5354	0.4646	Negligible	11
12	12	0	Correct	12
13	13	0	Correct	13
14	**19.3265**	**5.3265**	**Incorrect**	**19**
15	15	0	Correct	15

Mismatch is can be seen in Table 5.12 very easily which is shown in bold writing. Incorrect detection is belong to target value of 14. Equivalent word for this values is ‘tolga’ and mismatched with a value out of range.

5.2.2.4 Starting and Ending with Consonant Words Experiments

The fourth disyllabic word type is starting and ending with consonant words. Used words are listed in Table 5.13 and obtained accuracy performance results are listed in Table 5.14.

Table 5.13 Used disyllabic word voice samples in order.

Word	Target Value		Word	Target Value		Word	Target Value
balık	1		kaplan	6		sepet	11
demir	2		liman	7		şişman	12
fidan	3		mangal	8		tarak	13
günay	4		nermin	9		yağmur	14
japon	5		rahat	10		zemin	15

Table 5.14 Results of neural network experiments for disyllabic starting and ending with consonant words.

Wavelet	Order	Test 1	Test 2	Test 3	Test 4	Test 5	Average	Difference
Coiflets	3	93%	87%	87%	87%	87%	**88%**	7%
Reverse-Bio	2.4	87%	87%	87%	87%	93%	**88%**	7%
Symlet	5	87%	87%	87%	87%	93%	**88%**	7%
Discrete' Meyer	-	87%	87%	87%	87%	87%	**87%**	0%
Reverse-Bio	1.5	87%	87%	87%	87%	87%	**87%**	0%
Symlet	2	87%	87%	87%	87%	87%	**87%**	0%
Daubechies	2	87%	87%	87%	87%	60%	**81%**	27%
Daubechies	10	87%	87%	87%	60%	87%	**81%**	27%
Daubechies	6	87%	87%	87%	53%	87%	**80%**	33%
Symlet	8	87%	87%	87%	60%	73%	**79%**	27%
Coiflets	1	87%	93%	87%	87%	40%	**79%**	53%
Daubechies	1	93%	33%	93%	87%	87%	**79%**	60%
Biorthogonal	1.5	40%	87%	87%	73%	87%	**75%**	47%
Biorthogonal	3.9	40%	87%	67%	87%	87%	**73%**	47%
Biorthogonal	2.4	87%	67%	87%	87%	33%	**72%**	53%
Coiflets	5	93%	33%	93%	87%	47%	**71%**	60%
Reverse-Bio	3.3	47%	93%	33%	67%	40%	**56%**	60%
Overall Average		-	-	-	-	-	**79%**	30%

Related to result Table 5.14, third order Coiflets, fifth order Symlet and Reverse-Biorthogonal 2.4 wavelets achieve the best average accuracy performance value with 88%. The highest accuracy performance of tests is 93% which is obtained by various wavelets. 93% accuracy performance means fourteen correct and only one incorrect detections in total fifteen samples.

Reverse-Biorthogonal 1.5, second order Symlet and Discrete' Meyer wavelets achieved the most stable performance results with the same accuracy performance in all tests. Performance of Reverse-Biorthogonal 3.3 and third order Coiflets wavelet was the most unstable. Difference of their performance percentage achieved 60%.

One of the 93% accuracy performance belongs to the test that third order Coiflets wavelet used. For related test, correct detections can be seen on diagonal line in regression plot of neural network which is showed in Figure 5.11. Incorrect detection can be seen out of diagonal line in the same figure.

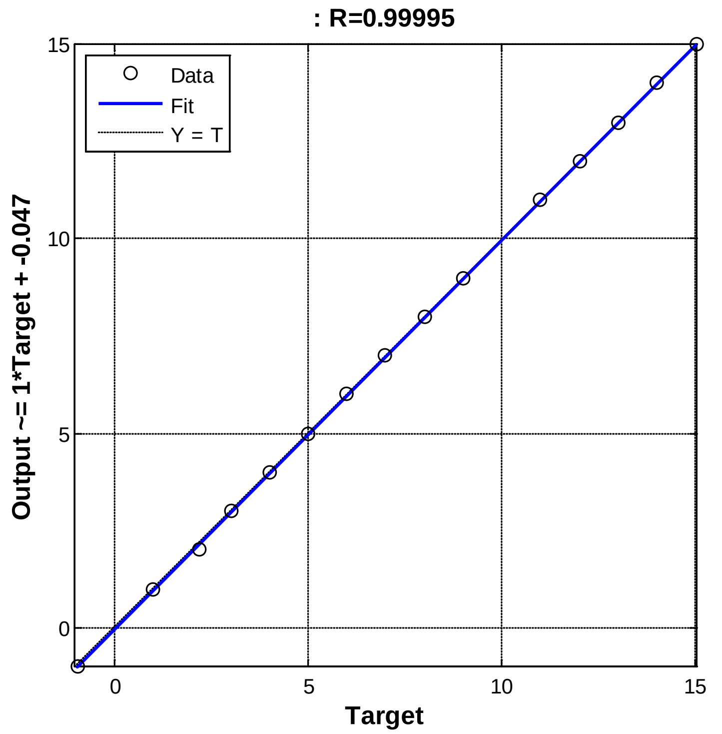

Figure 5.11 Regression plot of neural network.

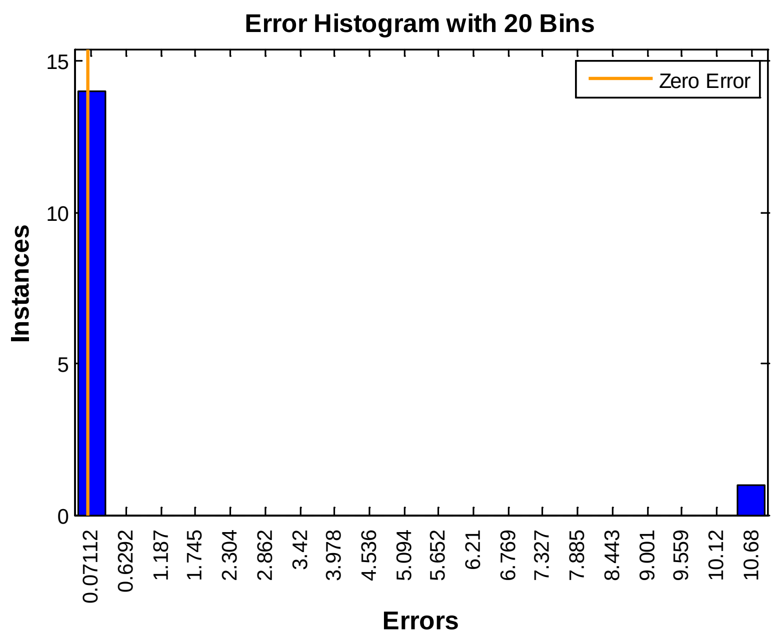

Figure 5.12 Error histogram plot of neural network.

Correct detections are accumulated around zero error line, one incorrect detection is accumulated at nearly eleven units away distance to zero line in the Figure 5.12.

Table 5.15 Comparison of target and result values.

Target Values	Output Values	Deviation	Explanation	Improved Result Values
1	1	0	Correct	1
2	2.2079	0.2079	Negligible	2
3	3	0	Correct	3
4	4	0	Correct	4
5	5	0	Correct	5
6	6	0	Correct	6
7	7	0	Correct	7
8	8	0	Correct	8
9	9	0	Correct	9
10	**-0.9545**	**10.9545**	**Incorrect**	**-1**
11	11	0	Correct	11
12	12	0	Correct	12
13	13	0	Correct	13
14	14	0	Correct	14
15	15	0	Correct	15

Mismatch is can be seen in Table 5.15 very easily which is shown in bold writing. Incorrect detection is belong to target value of 10. Equivalent word for this values is ‘rahat’ and mismatched with a value out of range.

5.2.3 Trisyllabic Word Based Neural Network Experiments

In these experiments, trisyllabic word voice samples are used as input data. Trisyllabic words are chosen indiscriminately. Used words are listed in Table 5.16 and obtained accuracy performance results are listed in Table 5.17.

Table 5.16 Used trisyllabic word voice samples in order.

Word	Target Value	Word	Target Value	Word	Target Value
araba	1	gereksiz	6	öğrenci	11
benzersiz	2	işçilik	7	sessizlik	12
cendere	3	karaman	8	şamata	13
çeşitli	4	mantıksız	9	ulaşım	14
denetim	5	nerede	10	yardımcı	15

Table 5.17 Results of neural network experiments for trisyllabic words.

Wavelet	Order	Test 1	Test 2	Test 3	Test 4	Test 5	Average	Difference
Coiflets	5	93%	87%	87%	87%	87%	**88%**	7%
Daubechies	2	87%	87%	93%	87%	87%	**88%**	7%
Discrete' Meyer	-	93%	87%	87%	87%	87%	**88%**	7%
Reverse-Bio	1.5	87%	87%	87%	87%	87%	**87%**	0%
Biorthogonal	1.5	87%	87%	93%	87%	60%	**83%**	33%
Reverse-Bio	2.4	67%	87%	87%	87%	87%	**83%**	20%
Reverse-Bio	3.3	40%	87%	87%	87%	87%	**77%**	47%
Coiflets	3	87%	87%	87%	87%	40%	**77%**	47%
Biorthogonal	2.4	80%	87%	87%	80%	47%	**76%**	40%
Symlet	2	53%	87%	87%	87%	67%	**76%**	33%
Biorthogonal	3.9	87%	87%	87%	47%	73%	**76%**	40%
Symlet	8	87%	33%	87%	87%	87%	**76%**	53%
Daubechies	1	80%	87%	53%	40%	87%	**69%**	47%
Daubechies	10	87%	87%	87%	47%	40%	**69%**	47%
Coiflets	1	60%	87%	73%	33%	87%	**68%**	53%
Symlet	5	87%	33%	40%	73%	87%	**64%**	53%
Daubechies	6	87%	47%	40%	93%	40%	**61%**	53%
Overall Average		-	-	-	-	-	**77%**	34%

Related to result Table 5.17, fifth order Coiflets, second order Daubechies and Discrete' Meyer wavelets achieve the best average accuracy performance value with 88%. The highest accuracy performance of tests is 93% which is obtained by various wavelets. 93% accuracy performance means fourteen correct and only one incorrect detections in total fifteen samples.

Reverse-Biorthogonal 1.5, second wavelet achieved the most stable performance results with the same accuracy performance in all tests. Performance of fifth and eighth Symlets, first order Coiflets and sixth order Daubechies wavelets were the most unstable. Difference of their performance percentage achieved 53%.

One of the 93% accuracy performance belongs to the test that fifth order Coiflets wavelet used. For related test, correct detections can be seen on diagonal line in regression plot of neural network which is showed in Figure 5.13. Incorrect detection can be seen out of diagonal line in the same figure.

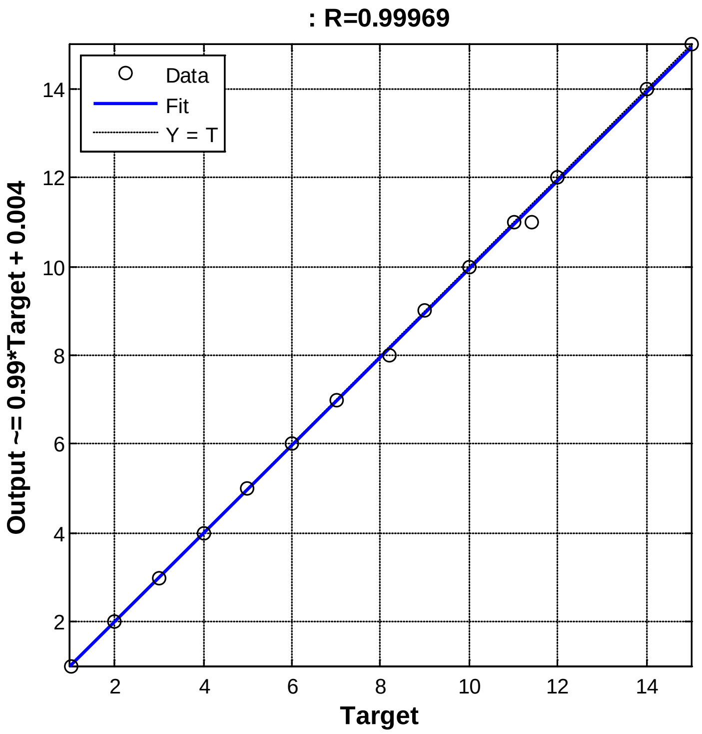

Figure 5.13 Regression plot of neural network.

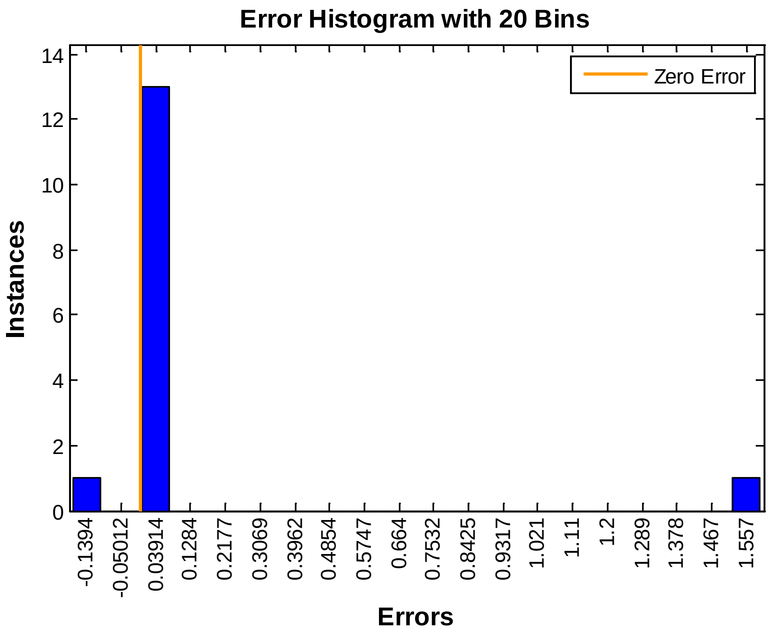

Figure 5.14 Error histogram plot of neural network.

Figure 5.12 shows the error histogram of the same test. Correct detections are accumulated around zero error line and one incorrect detection is accumulated at nearly one and a half unit away distance to zero line in the figure.

Mismatch is can be seen in Table 5.18 very easily which is shown in bold writing. Incorrect detection is belong to target value of 13. Equivalent word for this values is ‘şamata’ and mismatched with a ‘öğrenci’.

Table 5.18 Comparison of target and result values.

Target Values	Output Values	Deviation	Explanation	Improved Result Values
1	1	0	Correct	1
2	2	0	Correct	2
3	3	0	Correct	3
4	4	0	Correct	4
5	5	0	Correct	5
6	6	0	Correct	6
7	7	0	Correct	7
8	8.1840	0.1840	Negligible	8
9	9	0	Correct	9
10	10	0	Correct	10
11	11	0	Correct	11
12	12	0	Correct	12
13	**11.3988**	**1.6012**	**Incorrect**	**11**
14	14	0	Correct	14
15	15	0	Correct	15

5.2.4 Quadrisyllabic Word Based Neural Network Experiments

Quadrisyllabic words are chosen indiscriminately. Used words are listed in Table 5.19 and obtained accuracy performance results are listed in Table 5.20.

Table 5.19 Used quadrisyllabic word voice samples in order.

Word	Target Value	Word	Target Value	Word	Target Value
ayakkabı	1	denizaltı	6	kütüphane	11
bilgisayar	2	geleneksel	7	matematik	12
buzdolabı	3	iletişim	8	Pazartesi	13
çeşmealtı	4	kahverengi	9	şekerleme	14
çikolata	5	karşıyaka	10	yaratıcı	15

Table 5.20 Results of neural network experiments for quadrisyllabic words.

Wavelet	Order	Test 1	Test 2	Test 3	Test 4	Test 5	Average	Difference
Daubechies	10	87%	87%	87%	87%	93%	**88%**	7%
Biorthogonal	2.4	87%	87%	87%	87%	87%	**87%**	0%
Biorthogonal	3.9	87%	87%	87%	87%	87%	**87%**	0%
Discrete' Meyer	-	87%	87%	87%	87%	87%	**87%**	0%
Daubechies	1	87%	87%	87%	87%	47%	**79%**	40%
Symlet	8	87%	87%	87%	87%	40%	**77%**	47%
Coiflets	1	87%	87%	87%	33%	87%	**76%**	53%
Biorthogonal	1.5	93%	87%	87%	60%	47%	**75%**	47%
Symlet	2	80%	60%	87%	87%	53%	**73%**	33%
Daubechies	2	60%	87%	73%	87%	53%	**72%**	33%
Reverse-Bio	2.4	87%	87%	53%	87%	47%	**72%**	40%
Daubechies	6	87%	93%	47%	40%	87%	**71%**	53%
Reverse-Bio	3.3	87%	33%	47%	87%	87%	**68%**	53%
Coiflets	3	40%	53%	87%	40%	87%	**61%**	47%
Reverse-Bio	1.5	87%	87%	40%	53%	40%	**61%**	47%
Symlet	5	40%	67%	60%	87%	47%	**60%**	47%
Coiflets	5	73%	33%	40%	87%	53%	**57%**	53%
Overall Average		-	-	-	-	-	**74%**	35%

Related to result Table 5.20, tenth order Daubechies wavelet achieve the best average accuracy performance value with 88%. The highest accuracy performance of tests is 93% which is obtained by various wavelets. 93% accuracy performance means fourteen correct and only one incorrect detections in total fifteen samples.

Discrete' Meyer, Biorthogonal 2.4 and 3.9 wavelets achieved the most stable performance results with the same accuracy performance in all tests. Performance of Reverse- Biorthogonal 3.3, first and fifth order Coiflets and sixth order Daubechies wavelets were the most unstable. Difference of their performance percentage achieved 53%.

One of the 93% accuracy performance belongs to the test that tenth order Daubechies wavelet used. For related test, correct detections can be seen on diagonal line in regression plot of neural network which is showed in Figure 5.15. Incorrect detection can be seen out of diagonal line in the same figure.

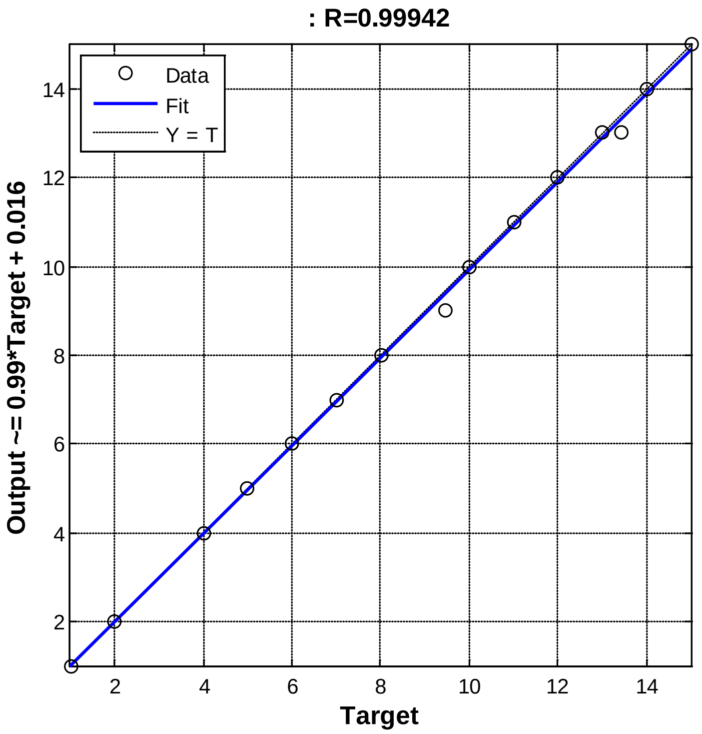

Figure 5.15 Regression plot of neural network.

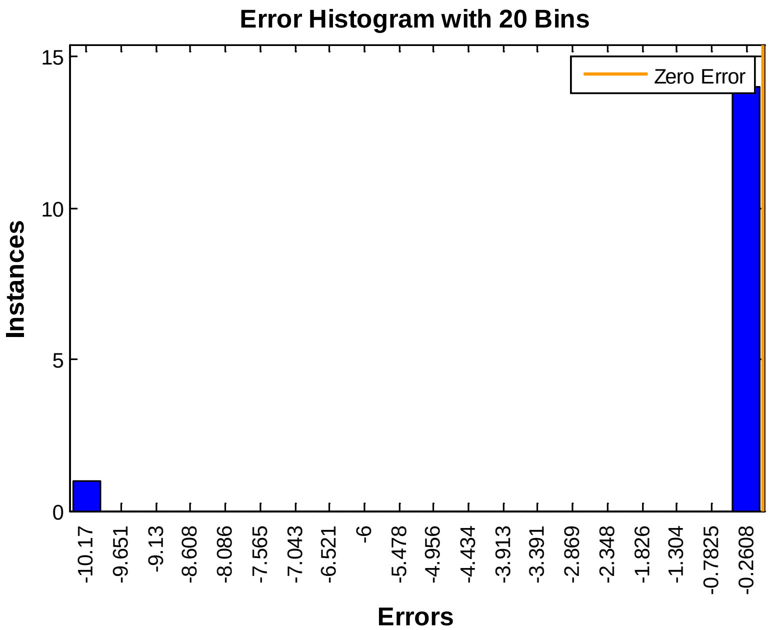

Figure 5.16 Error histogram plot of neural network.

Figure 5.16 shows the error histogram of the same test. Correct detections are accumulated around zero error line and one incorrect detection is accumulated at nearly one and a half unit away distance to zero line in the figure.

Table 5.21 Comparison of target and result values.

Target Values	Output Values	Deviation	Explanation	Improved Result Values
1	1	0	Correct	1
2	2	0	Correct	2
3	**13.4340**	**10.4340**	**Incorrect**	**13**
4	4	0	Correct	4
5	5	0	Correct	5
6	6	0	Correct	6
7	7	0	Correct	7
8	8	0	Correct	8
9	9.4494	0.4494	Negligible	9
10	10	0	Correct	10
11	11	0	Correct	11
12	12	0	Correct	12
13	13	0	Correct	13
14	14	0	Correct	14
15	15	0	Correct	15

Mismatch is can be seen in Table 5.21 very easily which is shown in bold writing. Incorrect detection is belong to target value of 3. Equivalent word for this values is ‘buzdolabı’ and mismatched with a ‘pazartesi’.

5.2.5 Mixed Words Based Neural Network Experiments

In these experiments, disyllabic, trisyllabic and quadrisyllabic word voice samples are chosen indiscriminately. Used words are listed in Table 5.22.

Table 5.22 Used word voice samples in order.

Word	Target Value		Word	Target Value		Word	Target Value
araba	1		irem	6		rahat	11
balık	2		kahverengi	7		şaka	12
denizaltı	3		liman	8		tolga	13
ece	4		ordu	9		ulaşım	14
gereksiz	5		övünç	10		yaratıcı	15

Table 5.23 Results of neural network experiments for mixed words.

Wavelet	Order	Test 1	Test 2	Test 3	Test 4	Test 5	Average	Difference
Daubechies	10	93%	93%	87%	87%	87%	**89%**	7%
Coiflets	3	93%	73%	87%	93%	87%	**87%**	20%
Biorthogonal	3.9	87%	87%	87%	87%	87%	**87%**	0%
Discrete' Meyer	-	87%	87%	87%	87%	87%	**87%**	0%
Reverse-Bio	2.4	87%	87%	87%	87%	67%	**83%**	20%
Symlet	8	87%	67%	87%	87%	87%	**83%**	20%
Coiflets	5	87%	87%	93%	87%	53%	**81%**	40%
Coiflets	1	87%	67%	67%	93%	87%	**80%**	27%
Biorthogonal	2.4	87%	87%	47%	87%	87%	**79%**	40%
Reverse-Bio	1.5	47%	87%	87%	87%	87%	**79%**	40%
Symlet	5	87%	87%	47%	87%	87%	**79%**	40%
Biorthogonal	1.5	87%	40%	87%	87%	87%	**77%**	47%
Daubechies	1	87%	87%	53%	67%	87%	**76%**	33%
Daubechies	6	87%	87%	53%	60%	60%	**69%**	33%
Daubechies	2	33%	87%	80%	60%	87%	**69%**	53%
Symlet	2	87%	40%	87%	40%	87%	**68%**	47%
Reverse-Bio	3.3	87%	47%	87%	87%	33%	**68%**	53%
Overall Average		-	-	-	-	-	**79%**	31%

Related to result Table 5.23 result values, tenth order Daubechies wavelet achieve the best average accuracy performance value with 87,9%. The highest accuracy performance of tests is 93% which is obtained by various wavelets. 93% accuracy performance means fourteen correct and only one incorrect detections in total fifteen samples.

Discrete' Meyer, Biorthogonal and 3.9 wavelets achieved the most stable performance results with the same accuracy performance in all tests. Performance of Reverse- Biorthogonal 3.3 was the most unstable. Difference of their performance percentage achieved 53,3%.

One of the 93% accuracy performance belongs to the test that tenth order Daubechies wavelet used. For related test, correct detections can be seen on diagonal line in regression plot of neural network which is showed in Figure 5.17. Incorrect detection can be seen out of diagonal line in the same figure.

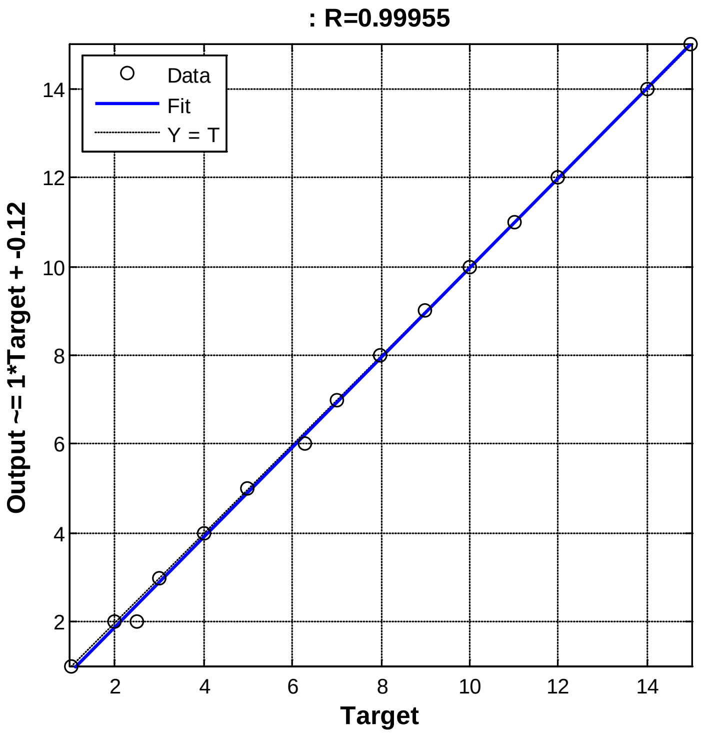

Figure 5.17 Regression plot of neural network.

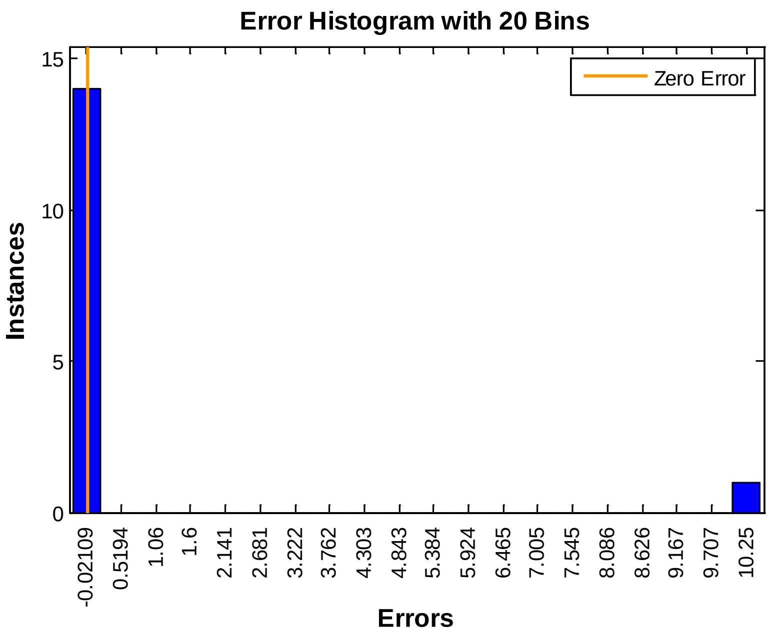

Figure 5.18 Error histogram plot of neural network.

Figure 5.18 shows the error histogram of the same test. Correct detections are accumulated around zero error line and one incorrect detection is accumulated at nearly ten units away distance to zero line in the figure.

Table 5.24 Comparison of target and result values.

Target Values	Output Values	Deviation	Explanation	Improved Result Values
1	1	0	Correct	1
2	2	0	Correct	2
3	3	0	Correct	3
4	4	0	Correct	4
5	5	0	Correct	5
6	6.2913	0.2913	Negligible	6
7	7	0	Correct	7
8	8	0	Correct	8
9	9	0	Correct	9
10	10	0	Correct	10
11	11	0	Correct	11
12	12	0	Correct	12
13	**2.4819**	**10.5181**	**Incorrect**	**2**
14	14	0	Correct	14
15	15	0	Correct	15

Mismatch is can be seen in Table 5.24 very easily which is shown in bold writing. Incorrect detection is belong to target value of 13. Equivalent word for this values is ‘tolga’ and mismatched with a ‘balık’.

CHAPTER SIX
CONCLUSION

6.1 Overview of Thesis

Aim of the thesis is to analyze the performance and efficiency of wavelet techniques on Turkish speech recognition. The feature matrix which is obtained from wavelet transform is used as input data for the neural network, then output data of the neural network is compared with input data and this comparison gives the performance of wavelet used in wavelet transform. For this purpose, different types of wavelets are used in experiments in order. All steps of experiments are realized in Matlab software.

6.2 Review of Experimental Results

To make a detailed analysis, lots of different Turkish words and word types are used in experiments. Different wavelets are used one by one with different Turkish word structures. Turkish words are mainly classified as monosyllabic, disyllabic, trisyllabic and quadrisyllabic words. By the way, voice of a letter can be changed related to its position in the word, so this must be considered in experiments also. That situation is tested in disyllabic words experiments and dissyllabic words are grouped by their letter sequence.

In all experiments, firstly fifteen words are read and recorded. Then selected seventeen different wavelets are used to create feature matrices. Each experiments include five tests and give an average accuracy percentage for used wavelet.

6.2.1 Review of Monosyllabic Word Based Experiments

Average performance of wavelets in this experiment is 83% so we can say that wavelet transform is a suitable for recognition of monosyllabic Turkish words. Best average performance belongs to third order Coiflets wavelet. Third order Coiflets wavelet obtained 91% average performance and its all result values are in stable.

Although the worst average performance belongs to Reverse-biorthogonal 3.3 and 1.5 wavelets, they were able to give 72% average performance, which can be considered as adequate performance.

6.2.2 Review of Disyllabic Word Based Experiments

In this experiment, disyllabic words are grouped in four classes. Their analysis and details are in following.

The first group includes starting and ending with vowel words. Average performance of wavelets in this experiment is 80% so we can say that wavelet transform is a suitable for recognition of this kind of disyllabic Turkish words. Best average performance belongs to tenth order Daubechies wavelet. Tenth order Daubechies wavelet obtained 89% average performance and its all result values are in stable. The worst average performance belongs to Discrete' Meyer wavelet with an average performance 68%. It’s not reliable but noticeable performance.

The second group includes starting with vowel and ending with consonant words. Average performance of wavelets in this experiment is 79% so we can say that wavelet transform is a suitable for recognition of this kind of disyllabic Turkish words. Best average performance belongs to third order Coiflets wavelet. It obtained 89% average performance and its all result values are in stable. The worst average performance belongs to fifth order Symlet wavelet with an average performance 64%. It’s not reliable but noticeable performance.

The third group includes starting with consonant and ending with vowel words. Average performance of wavelets in this experiment is 75% so we can say that wavelet transform is a suitable for recognition of this kind of disyllabic Turkish words. Best average performance belongs to fifth order Symlet wavelet. It obtained

89% average performance and its all result values are in stable. The worst average performance belongs to Biorthogonal 2.4 wavelet with an average performance 59%. It's not reliable performance.

The fourth group includes starting and ending with consonant words. Average performance of wavelets in this experiment is 79% so we can say that wavelet transform is a suitable for recognition of this kind of disyllabic Turkish words. Best average performance belongs to third order Coiflets, fifth order Symlet and Reverse-Biorthogonal 2.4 wavelets. They obtained 88% average performance and their all result values are in stable. The worst average performance belongs to Reverse-Biorthogonal 3.3 wavelet with an average performance 56%. It's not reliable.

6.2.3 Review of Trisyllabic Word Based Experiments

Average performance of wavelets in this experiment is 77% so we can say that wavelet transform is a suitable for recognition of trisyllabic Turkish words. Best average performance belongs fifth order Coiflets, second order Daubechies and Discrete' Meyer wavelets. They obtained 88% average performance and its all result values are in stable. The worst average performance belongs to sixth order Daubechies wavelet with an average performance 61%. It's not reliable performance.

6.2.4 Review of Quadrisyllabic Word Based Experiments

Average performance of wavelets in this experiment is 74% so we can say that wavelet transform is a suitable for recognition of quadrisyllabic Turkish words. Best average performance belongs tenth order Daubechies wavelet. It obtained 88% average performance and its all result values are in stable. The worst average performance belongs to fifth order Coiflets wavelet with an average performance 57%. It's not reliable performance.

6.2.5 Review of Mixed Words Based Experiments

Average performance of wavelets in this experiment is 79% so we can say that wavelet transform is a suitable for recognition of trisyllabic Turkish words. Best average performance belongs tenth order Daubechies wavelet. It obtained 89% average performance and its all result values are in stable. The worst average performance belongs Reverse- Biorthogonal 3.3 wavelet with an average performance 68%. It's not reliable but noticeable performance.

6.2.6 General Review of All Experiments

We can clearly say that wavelets are very good at recognition of Turkish voices. Experimental average recognition performance of all wavelets is 78%. I think this value proves the ability of discrimination of wavelets. In Figure 6.1, accuracy percentages of wavelets can be seen one by one.

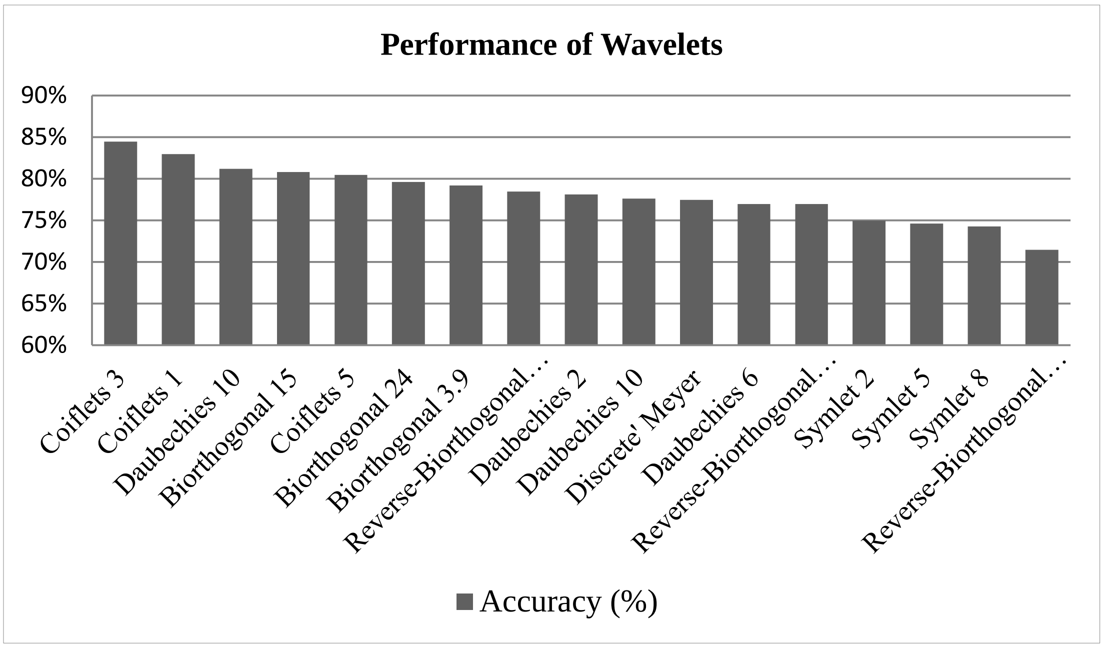

Figure 6.1 Performance of wavelets.

The best overall average performance is belongs to third order Coiflets wavelet with 84% achievement. Although the worst overall average performance is belongs to Reverse- Biorthogonal 3.3 wavelet, it has 71% achievement which can be considered successful.

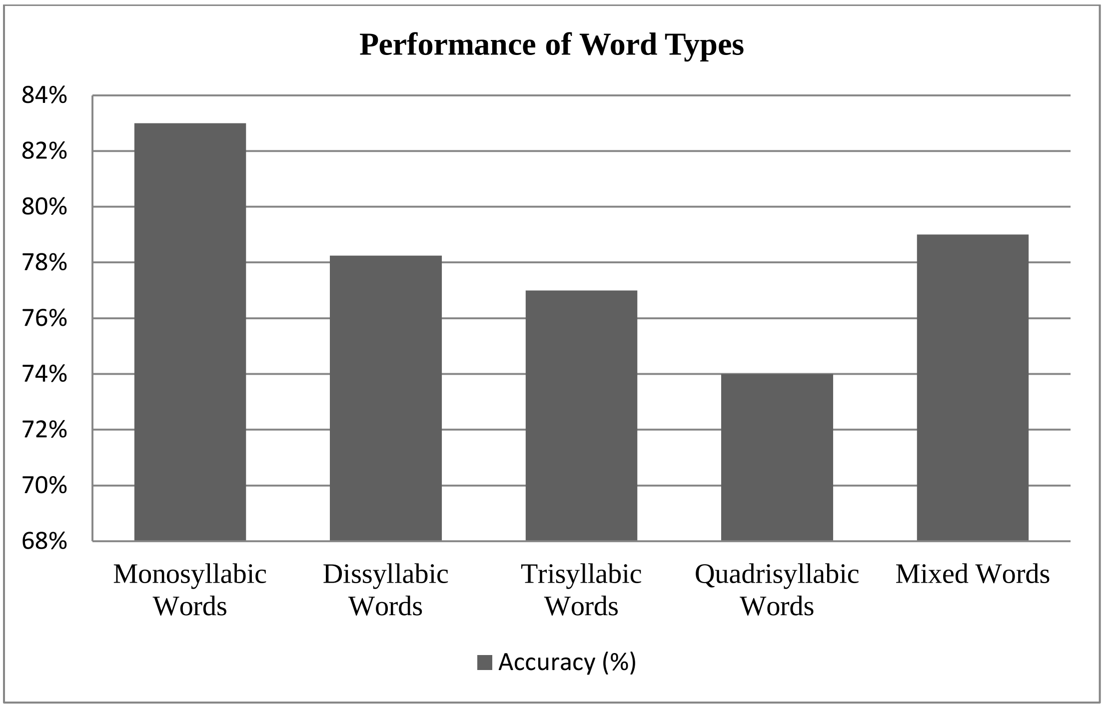

Figure 6.2 Performance of word types.

If you analyze the performance result related to syllable structures, you will see that wavelets shows best performance at shortest words. In some tests of monosyllabic words, 100% accuracy is obtained. Performance of mixed words is very close to average of performances of monosyllabic, disyllabic, trisyllabic and quadrisyllabic words.

6.3 Future Work

This thesis proved that wavelets are good at recognition of Turkish voice. In other words, wavelets are good at signal discrimination so new and very different application areas can be tested in the future. I want to analyze wavelets in different application areas in my doctorate instruction.

REFERENCES

Addison, P.S. (2012). *The illustrated wavelet transform handbook*. London : Institute of Physics Publishing.

Artificial Neural Network (n.d.). Retrieved May 15, 2014, from http://en.wikipedia.org/wiki/Neural_network

Beale, M.H., Hagan, M.T. & Demuth H.B. (2014) *Neural network toolbox user's guide™ R2014a.* Retrieved March, 25, 2014, from http://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf.

Demircan, Ö. (2009). *Türkçenin ses dizimi*. İstanbul : DER Yayınları.

Matlab R2014a Documentation Center (n.d.). Retrieved May 25, 2014, from http://www.mathworks.com/help/

Oppenheim, A.V., Willsky, A. S. & Nawab, S.H. (1997). *Signal & systems, (2nd ed.).* London : Prentice Hall.

Tebelskis, J. (1995). *Speech recognition using neural networks.* School of Computer Science Carnegie Mellon University.