AUTOMATIC SPIKE DETECTION USING FUZZY C-MEANS CLUSTERING

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

by
Zeynep Hilal İNAN

January, 2005
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"AUTOMATIC SPIKE DETECTION USING FUZZY C-MEANS CLUSTERING"** completed by **ZEYNEP HİLAL İNAN** under supervision of **ASSIST.PROF.DR.MEHMET KUNTALP** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist.Prof.Dr.Mehmet Kuntalp

Supervisor

Prof.Dr.Cüneyt Güzeliş

(Jury Member)

Assist.Prof.Dr.Adil Alpkoçak

(Jury Member)

Prof.Dr. Cahit HELVACI
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

First, I would like to thank my advisor, Asst. Prof. Dr. Mehmet Kuntalp for his continuous help and guidance throughout this work.

I am also very grateful to Dr. Nurettin Acır, for his support for the accomplishment of this thesis and besides, for the contribution of his studies to my knowledge.

Zeynep Hilal İNAN

AUTOMATIC SPIKE DETECTION USING FUZZY C-MEANS CLUSTERING

ABSTRACT

This thesis introduces a technique based on artificial neural networks (ANN) and fuzzy c-means clustering (FCM) method for detection of specific wave shapes in electroencephalogram (EEG) records which are used for diagnosis of epilepsy. Specific time domain features are extracted from each individual epileptiform wave in the records, and these features are fed as input to the detection process. A simple artificial neural network (ANN) model known as perceptron is used during the data pre-processing stage for a pre-classification. This pre-classification stage reduces data and eliminates definite spikes and definite non-spikes. The third group which consists of non-definite spikes and non-definite non-spikes are then processed by a fuzzy c-means classifier.

In the search for alternative methods, both raw and processed data are applied as input to the system. Also pre-classification stage removed to observe Fuzzy c-means classification performance and all results are compared to find out the performance of each approach.

The results of this study are evaluated by the sensitivity, specifity, selectivity and average detection rate criteria of the system which reveals the true-false detection rate of the process. Also the results are compared with other techniques priorily used by many researchers.

Keywords: EEG, Automatic spike detection, Fuzzy c-means clustering, Perceptron

FUZZY C-MEANS SINIFLANDIRMA İLE OTOMATİK DİKEN BELİRLEME

ÖZET

Bu tezde, yapay sinir ağları ve fuzzy c-means sınıflandırma yöntemi ile elektroensefalogram (EEG) kayıtlarından epilepsi hastalığının tanısında kullanılan belirli dalga şekillerinin otomatik olarak belirlenmesi için bir teknik tanıtılmaktadır. Kayıtlardaki epileptik şekle sahip sinyallerin zaman alanındaki belirli özellikleri çıkarılmış ve sisteme giriş olarak kullanılmıştır. Sinyallerin işlenmesi sürecinde kayıtlar önce perceptron olarak adlandırılan basit bir yapay sinir ağı sistemi ile sınıflandırmaya tabi tutulmuştur. Bu ilk sınıflandırma, sistemdeki veri yükünü azaltmakta ve belirgin olarak epileptik olan ve belirgin olarak epileptik olmayan dalgaları ayıklayarak epileptik olup olmadığı belirsiz üçüncü bir grup oluşturmaktadır. İşte bu üçüncü grup dalgalar fuzzy c-means sınıflandırma yöntemi ile epileptik veya epileptik olmayan şeklinde ayrılmak üzere işlenirler.

Alternatif yöntemlerin araştırılması amacıyla sisteme sinyallerden çıkarılan özelliklerin yanı sıra işlenmemiş sinyaller de girilerek performans ölçülmüştür. Ayrıca veriler ilk sınıflandırmaya tabi tutulmadan fuzzy c-means algoritması ile sınıflandırılmış ve sonuçlar karşılaştırılmıştır.

Bu çalışmanın sonuçları sistemin hassaslığı, belirleyiciliği, seçiciliği ve ortalama epileptik dalga yakalama derecesi ölçülerek hesaplanmıştır. Bu ölçütler sistemin doğru ya da yanlış yakalama oranını belirler. Ayrıca sonuçlar daha önce araştırmacılar tarafından kullanılan çeşitli tekniklerin sonuçları ile de karşılaştırılmıştır.

Anahtar sözcükler: EEG, Otomatik diken belirlenmesi, Fuzzy c-means sınıflandırma, Perceptron

CONTENTS

CHAPTER ONE
INTRODUCTION

1.1 Introduction

In this thesis, a technique based on fuzzy c-means clustering algorithm and artificial neural networks for detection of epileptiform waves in electroencephalogram (EEG) records is introduced. Electroencephalogram is a recording of electrical activity of the brain made by placing electrodes on the scalp. EEG signals are of great value for diagnosis of cerebral tumors, epilepsy and cerebral abscess and widely applied for diagnostic and research purposes (Snell, 2001). Epilepsy is defined as a sudden disturbance of the normal electrical activity of the brain and evaluation of electrical signals of brain carries high importance for the detection and classification of epilepsy.

The detection of epileptiform waves in EEG recordings is a subjective and time-consuming work for EEG specialists. As digital signal processing techniques involve and processing power of computers increase, research fields in biomedical engineering expand. Many researches are being conducted for automatization of epileptiform spike detection, and each has its own advantages and disadvantages. These researches differ in the utilization of input data and employed techniques.

Raw input data is the EEG recording acquired from clinical EEG monitoring systems. This data may be used as it is, but this requires an excessive amount of processing power since recordings are generally minutes long. Since epileptiform waves reveal themselves in forms of spikes, researchers generally choose to extract the individual spikes from the EEG recordings first. In some studies, these spikes are used as input to systems, while in some others specific features of these spikes are extracted and fed to the system. The latter method reduces the data load on the system and reduces processing time.

In this thesis, first individual spikes are extracted from the records, and then specific features of each spike are extracted by numerical techniques. These features include the amplitude and duration of the spike. As an alternative method, a reduced amount of raw data is also applied for the comparison of the results.

Then the spikes are pre-classified by two perceptrons fed by the extracted features. After this pre-classification, spikes are clustered into three groups which are definite epileptiform spikes, definite non-spikes, non-definite epileptiform spikes and non-definite non-spikes. (Acır, 2004)

This pre-classification stage greatly reduces amount of data to be processed and improves the system performance. (Acır, 2004) Third group of spikes are then fed to the second stage, i.e., the fuzzy c-means classifier.

Fuzzy c-means algorithm is becoming a popular method for classification purposes as it unites c-means clustering algorithm with fuzzy logic, which is very powerful in dealing with indefinite data and uncertainties in real life. (Inan, 2002)

The results of this study are evaluated by calculating the sensitivity, specifity, selectivity and average detection rate measures of the system. These parameters are a standard for EEG spike detection algorithms to help measure the true/false ratio of the system so that the system performances can be compared with each other.

Through out this study, different combinations of input data and minor differentiations in the techniques are tried. Feeding processed and raw data to fuzzy c-means classifier is tried and the results are compared with each other. Effects of including pre-classification stage are also investigated by removing this stage from the system.

In this thesis, first an introduction to epilepsy, electroencephalography and detection of spikes in the electroencephalogram is made. In the third chapter the basics of neural networks are mentioned and the perceptron model is explained. The fourth chapter is dedicated to Fuzzy logic and fuzzy control. Details of clustering techniques and fuzzy c-means classification are presented in the fifth chapter.

In the sixth chapter, the application is introduced and the results of this study are given. The last chapter is the conclusion part.

CHAPTER TWO
EPILEPSY AND ELECTROENCEPHALOGRAM

2.1 Epilepsy

Epilepsy is a very common neurological disorder. It may be defined as a symptom where a sudden and transient disturbance occurs in normal electrical activity of the brain. It ceases spontaneously and usually recurs, and in most typical form of the disease it's accompanied by seizures. 4-5% of the population suffer epilepsy some time in their life and 1% have chronic epilepsy (Betts, 1998).

Epilepsy is a sudden and synchronuous discharge of brain cells. It may appear because of hereditary (genetic) nature. Also cerebral tumors and traumatic reasons that result in scarring of cerebral cortex may result in epilepsy.

Symptoms depend on the location of seizure onset within the brain and the spread of the onset. Epilepsy is characterized by sudden and transient disturbances of mental function and/or movements of the body that result from excessive discharging of groups of brain cells. (Kiloh, McComas, Osselton, & Upton A, 1981) (Niedermeyer, & Silva, 1993).

There are many seizure types, but mainly they are divided into two main groups: partial and generalized. Partial seizures affect a part of the brain whereas generalized seizures involve the entire brain and usually come with unconsciousness. (Holmes, 1997)

Electroencephalogram recordings and epileptiform spike detection is very important in diagnosis of epilepsy. Therefore many researches are being conducted in the automated epileptiform spike detection area.

Epileptic Seizures may be classified as below:

1. Partial (focal, local) seizures
   (a) Simple partial seizures (consciousness not impaired)
      i. With motor signs
      ii. With somatosensory or special sensory symptoms
      iii. With autonomic symptoms or signs
      iv. With psychic symptoms, rarely occurring without impairment of consciousness and more commonly experienced as complex partial seizures
   (b) Complex Partial Seizures (with impairment of consciousness)
      i. Simple partial onset, followed by impairment of consciousness
      ii. With impairment of consciousness at onset
   (c) Partial seizures evolving into generalized seizures (tonic-clonic,tonic,or clonic)
      i. Simple partial seizures evolving to generalized seizures
      ii. Complex partial seizures evolving to generalized seizures
      iii. Simple partial seizures evolving to complex partial seizures evolving to generalized seizures
2. Generalized seizures (convulsive or non-convulsive)
   (a) Absence seizures
      i. Typical absences
      ii. Atypical absences
   (b) Myoclonic seizures (myoclonic jerks, single or multiple)
   (c) Clonic seizures
   (d) Tonic seizures
   (e) Tonic-clonic seizures
   (f) Atonic seizures (astatic)
3. Unclassified epileptic seizures

2.2 Electroencephalogram (EEG)

EEG recordings are very important in study of nervous system since brain activity is measured during electroencephalogram. These recordings are used for diagnosis of many abnormalities resulting from the brain; from epilepsy to alcoholism. EEG measures the electric potentials caused by neurons and these potentials have generally temporal resolutions of a few ms.

2.2.1 Origins of EEG and Biological Background

EEG is a time varying record of the variations of the electrical activity of large ensembles of neurons. A neuron is the elementary nerve cell and it's the fundamental building block of the biological neural network.

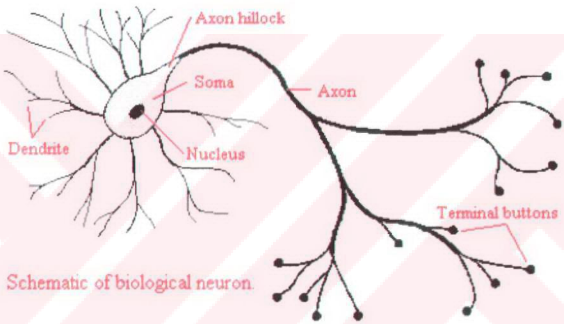


Figure 2.1 The schematic representation of a neuron (Fraser, 1998)

As shown in figure 2.1, a neuron has three major parts: cell body or so called soma, axon and dendrites. Dendrites receive information from neurons through axons. Axon is a long cylindrical connection that carries impulses from the neuron.

Each branch of axon connects to dendrites of other neurons through small gaps called synapse. Synapse is the place where neuron introduces its signal to neighboring neurons. These signals are electrical impulses. A synapse converts a pre-synaptic electrical signal into chemical signals and re-converts into electrical signal at its output. It can impose excitation or inhibition to the receptive neuron.

The receptive neuron is able to respond to the total of its inputs aggregated within a short time called the period of latent summation. The neuron's response is generated if the total potential of its membrane reaches a certain level. Neuron produces an impulse and sends it to its axon if this threshold is reached.

Incoming impulses may be excitatory or inhibitory and synapse determines this behavior. That's why positive or negative weights can be assigned to these connections. This biological process is also the foundation of artificial neural networks.

The resting membrane potential of a neuron is about -70 mV. If the excitatory inputs exceed the inhibitory inputs, and if the depolarization reaches -55mV, neuron sends information down an axon, away from the cell body. This is called an action potential. For any given neuron, the size of the action potential is always the same.

EEG signal is the recording of the changes in the action potentials. It depends on the operations of neuronal ensembles rather than on the actions of single neurons. This is because the amplitude of the potentials of single neurons is too small to be detected by electrodes.

The major sources of the EEG are the collective postsynaptic activity in the cell bodies and dendrites located in the cortical layers under the recording electrode. Action potentials contribute little to surface potentials except when there are synchronous action potentials in a large number of neurons. Synaptic potentials contribute more to the EEG because they are slower than action potentials and thus can summate.

Identifiable features of the EEG can be used to differentiate between the brain states of the patient, for example, during sleep, quiet wakefulness, etc.

2.2.2 Recording the EEG

Any recording system for the EEG requires the following components:

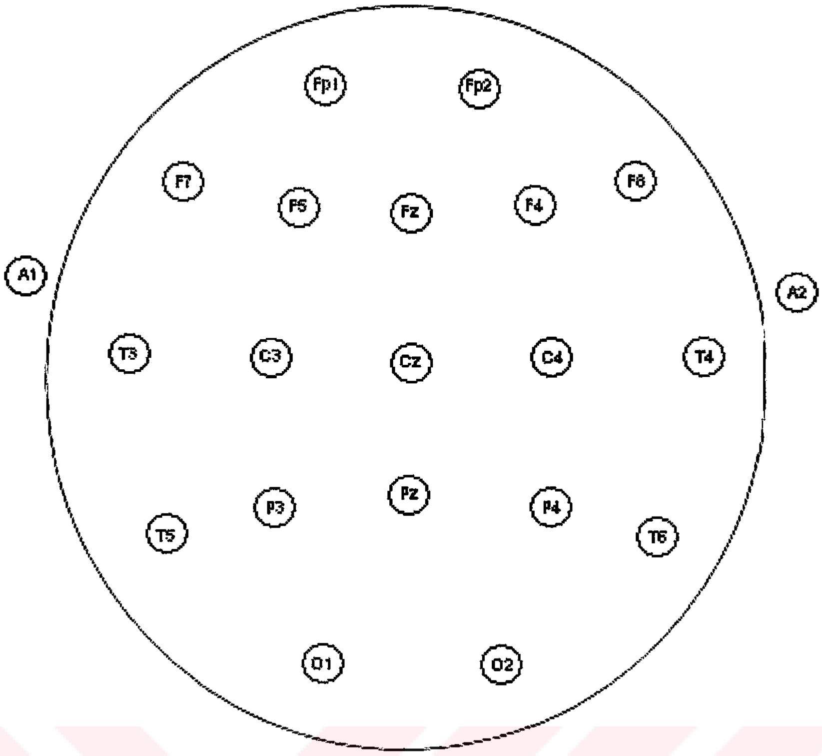
- Electrodes: Attached to the scalp. They gather the electrical signals at the surface of the scalp.

- Amplifiers: Amplitude of the EEG signals measured at the electrodes is very low. (10-100μV). Amplifiers are used to strengthen the signals.

- Filters: Utilized to remove unwanted noise from the EEG signal before recording. This noise may include power lines, radio stations or biological signals.

- Recording unit: Records the EEG signal. The temporal EEG data is stored in digital environment.

A number of different configurations have been proposed for electrode placement but the 10-20 International System is now the most common layout in use. The electrode positions and labels for the 10-20 system are shown in figure 2.3.



FP Pre-Frontal
F Frontal
C Central
T Temporal
P Parietal
O Occipital
A Ear

Figure 2.2 10-20 International electrode placement.(Jasper, 1958)

The 10-20 system is based on the relationship between the location of an electrode and the underlying area of cerebral cortex. Even numbers (2, 4, 6, 8) refer to the right hemisphere and odd numbers (1, 3, 5, 7) refer to the left hemisphere. The z refers to an electrode placed on the midline. Also note that the smaller the number, the closer the position is to the midline. The "10" and ""20" (10-20 system) refer to the 10% and 20% inter electrode distance. (Jasper, 1958)

The recorded signals making up the EEG are the potential differences between pairs of electrodes. There are two recording approaches used:

Referential: Potential difference measured between an electrode on the scalp and a reference electrode attached elsewhere.

Bipolar: Potential difference measured between two electrodes.

Unfortunately the recording of brain activity is easily disturbed. Any signal activity which does not arise from the brain is considered to be a recording artifact. There are many possible causes of artifact including muscle activity (e.g., scalp muscles or eye motion), mains electricity interference and patient and electrode movement.

2.2.3 Acquisition of EEG signals

The EEG process begins with placing a large number of electrodes on a patient's scalp which detect electrical signals that are given off by nerve cells in the brain. One of them acts as a reference electrode. Hence, each *channel* in an EEG system represents the potential difference between two electrodes, with most of the channels sharing a common reference electrode.

The placement of electrodes is very important. They are placed over the frontal, parietal, occipital, and temporal lobes according to a conventional scheme (Jasper, 1958). In addition, the electrodes must not be too close, otherwise the EEG will look like a straight line instead of showing the brainwaves.

2.2.4 Amplification of EEG signals

The EEG signals are very weak (20-100 microvolts). Therefore the signals are sent to amplifiers where they are amplified 10,000 times. At the front end is a high gain pre-amplifier with a high input impedance so as not to draw current from the scalp and distort the measurements. The pre-amplifier output is then further conditioned and amplified to meet the specific requirements of the ADC. The user selects the bandwidths of interest, typically a sub kilohertz range, and the system sets the necessary filters and sampling rates for digital acquisition.

2.2.5 The Noise in the EEG signal

The noise in an EEG system is dominated by biological signals. In addition, strong radio stations and power-line frequencies can couple into the EEG leads. The movements of the electrodes also add noise to the system. It is thus important that the electrodes remain stationary. An adhesive is used to hold them in place.

2.2.6 Interpretation of EEG waveforms

EEG signals can always be recorded from the living brain, even during sleep and unconscious states. The waveforms obtained vary depending on the degree of activity of the cerebral cortex. Often the waveforms appear irregular, but sometimes distinct patterns can be observed on the basis of the wave's amplitude and frequency. The most widely accepted indication of brain death is electro cerebral silence - an essentially flat EEG. The analysis of EEG recordings is the primary method of diagnosis in epilepsy.

Unfortunately the recording of brain activity is easily disturbed. Any signal activity which does not arise from the brain is considered to be a recording artifact. There are many possible causes of artifact including muscle activity (e.g., scalp muscles or eye motion), mains electricity interference and electrode movement.

For automated epileptiform spike detection from eeg recordings, several features derived from the EEG may be used. As mentioned before, presenting raw input data to the detector is not practical. It requires excessive processing time and power. So researchers generally look for a set of extracted features and/or try to reduce amount of data to be processed. These features may be in frequency domain and/or time domain.

Examples of frequency domain parameters are reflection coefficients from linear prediction which is used previously for eeg analysis and total signal power (Pardey, Roberts, & Tarassenko, 1996). Reflection coefficients will not indicate changes in signal amplitude, but the total signal power will show these differences.

Time domain parameters generally include amplitude, duration of first and second half waves of the spike and time derivative. Time derivative may be used in detection of the peak of the spike.

It is difficult to compare results from different papers reporting different methods but there is a set of standard statistical measures in addition to accuracy. In presenting our results, four of these measures are used: *sensitivity*, *specifity*, *selectivity* and *average detection rate (ADR)* (Pang, Upton, Shine, & Kamath, 2003). These are defined as:

$$\text{Sensitivity} = \frac{TP}{TP + FN} \times 100 \qquad [2.1]$$

$$\text{Specifity} = \frac{TN}{TN + FP} \times 100 \qquad [2.2]$$

$$\text{Selectivity} = \frac{TP}{TP + FP} \times 100 \qquad [2.3]$$

$$\text{Average Detection Rate} = \frac{Sensitivity + Specifity}{2} \times 100 \qquad [2.4]$$

where,

TP - True Positives: Number of epileptiform spikes detected

TN - True Negatives: Number of non-epileptiform activity detected

FP - False Positives: Number of activity incorrectly labeled as epileptiform

FN - False Negatives: Number of spikes incorrectly labeled as non-epileptiform

2.2.7 EEG Waveforms

Brain electrical activity consists primarily of rhythms in different frequency bands which have standard names. In a wakeful adult subject the alpha rhythm dominates but this change with brain state (thus in deep sleep the dominant frequency is much lower).

EEG signals are analyzed in the temporal and spatial domains. Analysis of the frequency components of the EEG is based on principles developed by Fourier. Within a relatively short time interval, a few dominant frequency bands are observed:

Table 2.1: Standard EEG frequency bands

Name	**Frequency Range**
Delta	less than 4Hz
theta	4-8Hz
alpha	8-13Hz
beta	greater than 13Hz

<u>Alpha waves</u> are associated with a state of relaxed wakefulness; they are recorded best over the parietal and occipital lobes.

<u>Beta waves</u> are seen during intense mental activity; they are recorded over the frontal regions.

<u>Delta and theta waves are</u> both associated with sleep; they have the largest amplitudes of EEG activity.

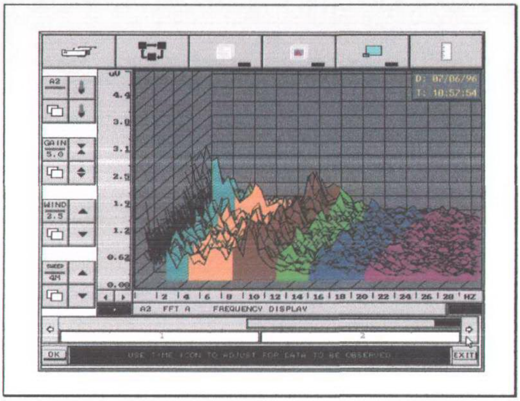


Figure 2.3 Example of EEG record

2.2.8 Abnormalities in EEG

A number of different abnormal patterns may be seen in the EEG of an awake subject. These are:

- Slow waves: Brain waves are collected in specific frequency bands. Waves with lower frequency should not be observed.
- Spikes/Sharp waves: These are similar in appearance but of different durations. Spikes last less than 70ms, sharp waves have a typical duration in the range 70-200ms.
- Spike and wave: A combination of spike and slow waves. This pattern lasts between 150-300 ms and repeats at a rate of approximately 3Hz.
- Depression: A period where signal amplitude is decreased.

The abnormal patterns seen in epileptic patients must be evaluated in two categories: ictal and inter-ictal. During the ictal period, patient is having an epileptic seizure. Between the seizures is the inter-ictal period.

It's important to detect the abnormalities in the eeg in the inter-ictal period and the most common form of inter-ictal abnormality is spiking (in the form of individual spikes, spike trains or spike and wave features). These spikes are seen in the majority of patients with epilepsy, whereas only a very small number of non-epileptic patients show this feature. For this reason inter-ictal spike detection plays a large part in the diagnosis of epilepsy.

A very different EEG pattern is seen during the ictal period consisting of high amplitude synchronized periodic waveforms, reflecting abnormal discharge of a large group of neurons. Although inter-ictal spikes offer evidence of epilepsy this can only be confirmed, and a complete diagnosis made, by an observed seizure.

An epileptiform spike generally has a duration between 20ms and 70ms, and the apex of the spike is sharp (Acır, 2004). Spike may be seen alone but it may also be followed by a slow wave. In this case the signal is called a 'spike and slow wave complex' and the duration is between 150 to 350 ms.

2.2.9 Medical Applications of the EEG

2.2.9.1 Epilepsy:

The EEG is the leading test used to help diagnose epilepsy. Large populations of neurons are activated synchronously during an epileptic seizure. Many people who suffer from seizures do not have detectable brain lesions, and tests like the MRI and CT scan show only normal brain structure. However, the EEG can show abnormal electrical function of the brain even when other tests are normal. There are two categories of EEG readings for diagnosis of epilepsy:

- inter-ictal EEG and
- ictal EEG

The Inter-ictal EEG: Interictal *EEG* is a recording taken when a patient is not having seizures. This reading is done to look for abnormal activity that can occur in a patient with epilepsy in the absence of an actual seizure. In the interictal EEG recording, a spike is usually seen. The spike denotes an explosion of electrical activity occurring inside the brain. These spikes can be confined to one area of the brain (focal), or many areas (multifocal). The localization of the spikes can determine the type of epilepsy a person has and which medications to use for their treatment.

*The Ictal EEG: T*he ictal EEG is recorded during a seizure. This reading is done when a patient's seizures fail to respond to treatment. The ictal EEG recordings can show rhythmic activity, but it can also show other EEG patterns depending on whether the recording was done with scalp electrodes or implanted electrodes.

A person with generalized epilepsy can show a recording of widespread brain involvement. A person with focal epilepsy will usually show a seizure beginning at a specific area of the brain and then becoming widespread.

Electroencephalograms are complex signals of massive volume that require expert personal judgment for analysis. It is important that epileptiform discharges be detected so that patients may be accurately and effectively diagnosed with and treated for epilepsy.

2.2.9.2 Insomnia:

Sleep deprivation is one of the most common disorders of people today. Bright light exposure is a common treatment for many sleep difficulties, especially those associated with alterations in the circadian timing system. Electroencephalograms have been used to research bright light therapy and its effectiveness. EEG is also used to distinguish various stages of sleep.

2.2.9.3 Stroke:

Medical history is filled with accounts of patients having stroke-like symptoms. In the past, it was difficult to detect abnormalities inside the brain without resorting to surgery. The electroencephalogram, with its ability to detect abnormalities, such as those resulting from strokes, has opened many doors for the detection and treatment of strokes.

2.2.9.4 Alcoholism:

As one of the most readily available and widely used drugs in society, alcohol has been a focus of many researchers. Questions about the effects of alcohol on the body, particularly the brain, as well as inquires about alcoholism as a disease are continually posed by scientists. It is known that alcohol misuse has been linked to a large number of deleterious effects on the central nervous system. Alcoholic patients may be affected with several disorders at one time, making it difficult to identify the cause of cognitive dysfunction in chronic alcoholics.

Electroencephalography has become a major tool in aiding physicians to diagnose many of these disorders, as well as helping researchers to look closer at the origins of alcoholism, along with its effects.

2.3 Necessity for an automated spike detector

The main reasons for an automated analysis include:

- Subjective nature of EEG analysis. There is a possibility of different readers having different conclusions.
- Time consuming EEG evaluation because of long recordings.

In this project, Automatic spike detection for diagnosis of epilepsy will be performed by a clustering algorithm called fuzzy c-means (also known as fuzzy k-means).

CHAPTER THREE
ARTIFICIAL NEURAL NETWORKS AND PERCEPTRON

Artificial neural networks (ANN) are parallel distributed processing models. Neural networks are composed of simple elements operating in parallel and information processing takes place through the interaction between a large numbers of these elements. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. We can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements. (Inan, 2002)

In order to understand the operations of the human brain, many researchers have chosen to create models based on findings in biological neural research. Neurobiologists have shown that human nerve cells have no memory of their own; rather memory is made up of a series of connections, or associations, throughout the nervous system. A response in one cell triggers responses in others which trigger more in others until these responses eventually lead to a so-called recognition of the input stimuli. This is how we recognize objects or events that we see.

3.1 Characteristics of ANN

Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. The network is adjusted, based on a comparison of output and the target, until the network output matches the target. Typically many such input/target pairs are used, (in supervised type of learning) to train a network.

Main features of artificial neural networks include learning, generalization and parallelism.

Learning: Neural networks has the ability to learn, i.e., for the desired task that will be performed by network, some input-output samples are given to the network and the network modifies its weights in a specific way according to these samples. That's why learning corresponds to parameter changes. This procedure is called the training of network. Through learning, the network generates its own rules for the function to be performed and can give maybe not exact, but sufficiently accurate responses. There exist three types of training methods which are supervised learning, reinforcement learning and unsupervised learning. The aim of the training procedure is to minimize the error between the desired and actual outputs. Training stops when the error is below a specified value or when the weights do not change any further.

Generalization: The network generates its own rules according to input-output samples presented to network in training through modification of weights. Neural networks are capable of giving accurate responses to previously unseen inputs, i.e. , the network generalizes the input-output relation that it has learned through training.

Parallelism: A neural network is a parallel distributed processing model, meaning that, all the input data is presented to network at the same time, and each neuron in the network is responsible of processing a small amount of data. Parallelism also yields three other important properties of neural networks: speed, fault-tolerant structure and efficiency.

3.2 Implementation Areas

Neural networks have been used to perform complex functions in various fields of application including pattern recognition, identification, classification, speech, vision and control systems.

The supervised training methods are commonly used, but other networks can be obtained from unsupervised training techniques or from direct design methods. Unsupervised networks can be used, for instance, to identify groups of data. Certain kinds of linear networks and Hopfield networks are designed Neural Network including connections (called weights) between neurons Input Output Target

3.3 Neural Network Model

In a large network the contribution of a single weight is often slight; it's the effect of combinations of connection weights that determines the output. The process of training a network is that of finding a set of values for the weights that make the network do what you want it to do.

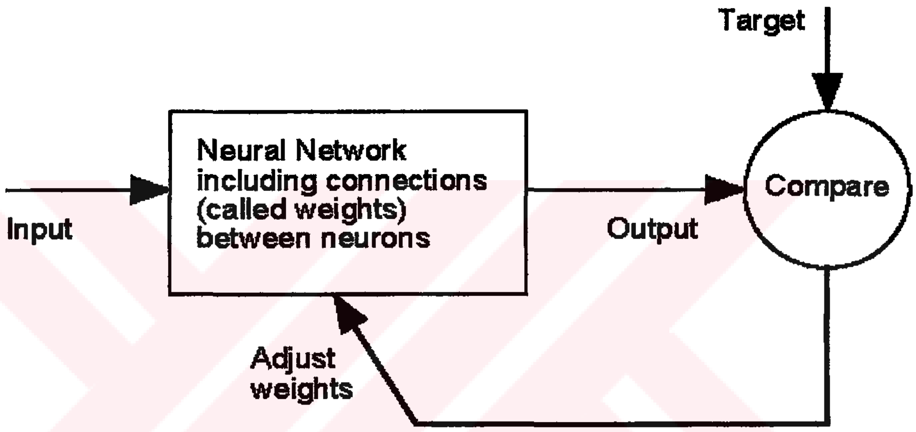


Figure 3.1 Neural network model

Every neuron model consists of a processing element with synaptic input connections and a single output. It sums the incoming signals, and then the signals are passed through a transfer function which is generally nonlinear. The output signal is fed to other neurons. The symbolic representation below shows a set of weights and the neuron's processing unit, or node.

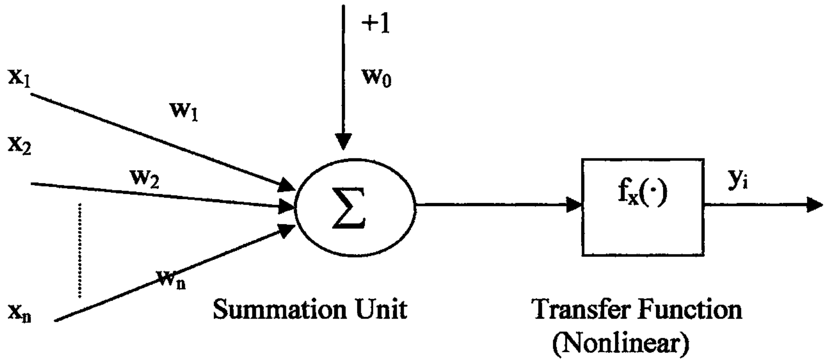


Figure 3.2 A single neuron

In the figure 3.2, $x_1, x_2, \ldots x_n$ are the inputs of the network, $w_1, w_2, \ldots w_n$ are the connection weights, w_0 is the weight determining the threshold, and $f_x(\cdot)$ is the transfer function. We can formulate the neuron's function as follows:

$$y = f\left(\sum_{i=1}^{n} \bar{\omega}_i^{\,T} \bar{x}_i\right) \qquad [3.1]$$

The function f is also referred to as activation function. Different types of activation functions can be utilized. Neural networks differ from each other in their learning rules. There are a variety of learning rules that establish when and how the connecting weights change.

Neural networks have been used to perform complex functions in various fields of application including pattern recognition, identification, classification, speech, vision and control systems.

The supervised training methods are commonly used, but other networks can be obtained from unsupervised training techniques or from direct design methods. Unsupervised networks can be used, for instance, to identify groups of data. Certain kinds of linear networks and Hopfield networks are designed Neural Network including connections (called weights) between neurons Input Output Target.

3.4 Perceptron

A perceptron is a simple element simulating the biological neuron. Perceptrons may be connected with each other in forms of layers to form a multilayer perceptron that simulates an associative memory. A perceptron is composed of an input layer and output layer of nodes, each of which are fully connected to the other. Assigned to each connection is a weight which can be adjusted so that, given a set of inputs to the network, the associated connections will produce a desired output. The adjusting of weights to produce a particular output is called the "training" of the network which is the mechanism that allows the network to learn. Perceptrons are among the earliest and most basic models of artificial neural networks, yet they are at work in many of today's complex neural net applications.

By adjusting the weights on the connections between layers, the perceptron's output could be "trained" to match a desired output. Training is accomplished by sending a given set of inputs through the network and comparing the results with a set of target outputs. If there is a difference between the actual and the target outputs, the weights are adjusted on the adaptive layer to produce a set of outputs closer to the target values.

New weights are determined by adding an error correction value to the old weight. The amount of the correction is determined by multiplying the difference between the actual output ($x[j]$) and target ($d[j]$) values by a learning rate constant (η). If the input node's output ($y[i]$) is a 1, that connection weight is adjusted, and if it sends 0, it has no bearing on the output and subsequently, there is no need for adjustment.

This training procedure is repeated until the network's performance no longer improves. The network is then said to have "converged". At this point, it has either successfully learned the training set or it has failed to learn all of the answers correctly (Dayhoff, 1990). If it is successful, it can then be given new sets of input and generally produce correct results on its own.

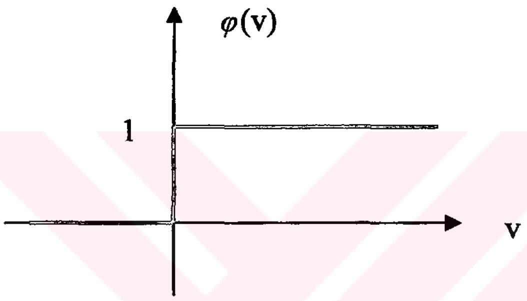
3.4.1 Transfer functions

A neuron's function is stated as:

$$y = f(\sum_{i=1}^{n} \bar{\omega}_i^T \bar{x}_i) \qquad [3.2]$$

where f is referred to as activation function. For a perceptron, activation functions may take the following forms:

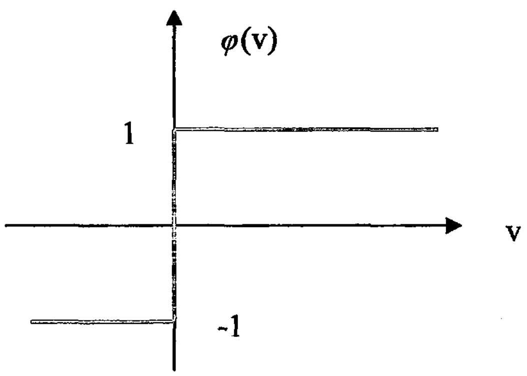
1. Unipolar or hard-limiter Threshold function:



Unit Step Function

$$\varphi(v) = \begin{cases} 1, & v \geq 0 \\ 0, & v < 0 \end{cases} \qquad [3.3]$$

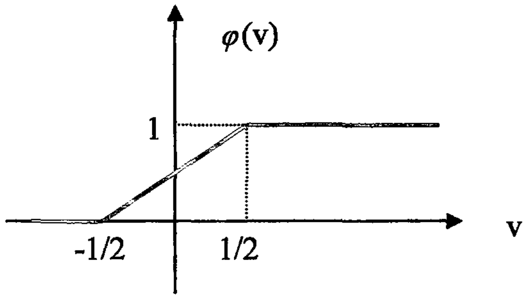
2. Bipolar Threshold function(Also known as Threshold logic unit – TLU):



Signum Function

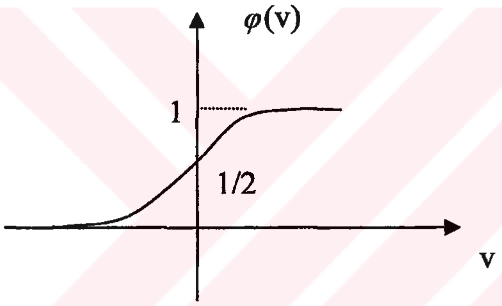
$$\varphi(v) = \begin{cases} 1, & v \geq 0 \\ -1, & v < 0 \end{cases} \qquad [3.4]$$

3. Unipolar piecewise Linear Sigmoid:



$$\varphi(v) = \begin{cases} 1, & \tfrac{1}{2} \leq v \\ v+\tfrac{1}{2}, & -\tfrac{1}{2} < v < \tfrac{1}{2} \\ 0, & v \leq -\tfrac{1}{2} \end{cases} \quad [3.5]$$

4. Unipolar Continuous sigmoid:



$$\varphi(v) = \frac{1}{1+e^{-\lambda V}} \quad (\lambda>0) \qquad [3.6]$$

A perceptron with bipolar threshold function is called a discrete perceptron while a perceptron with bipolar continuous sigmoid activation function is called continuous perceptron.

3.4.2 Perceptron Learning Rule

For the perceptron learning rule, learning signal is the difference between the desired response and actual neurons response. This kind of learning is supervised and weights are adjusted according to the following formula:

$$w[k+1] = w[k] + \eta(d^S - y^S)\mathbf{x}^S \qquad [3.7]$$

where *w[k]* is current values of the weights, *w[k+1]* is the updated value of weights, η is the learning rate constant, *d* is desired output, *y* is actual neuron output and *x* is the input vector. This formula provides that weights are updated in the way to reduce the difference between the desired and actual output. Actual output of perceptron can be adjusted using

$$y^S = \varphi(\mathbf{w}^{\mathbf{T}}\mathbf{x}^{\mathbf{s}} - \tau) \qquad [3.8]$$

where τ is the threshold value and φ is the activation function.

In this thesis, activation function for the perceptron is selected as bipolar threshold function also known as hard-limiter or signum function. According to this function the perceptron output becomes:

$$y^S = \mathrm{sign}\,(\mathbf{w}^{\mathbf{T}}\mathbf{x}^{\mathbf{s}} - \tau) \qquad \mathrm{sign}(v) = \begin{cases} 1, & v \geq 0 \\ -1, & v < 0 \end{cases} \qquad [3.9]$$

During the training process, if the input set is geometrically linearly separable, training algorithm stops at a weight vector providing a separating hyperplane. This means perceptron learning rule found a weight vector defining the normal and the offset of a hyperplane in an n-dimensional space.

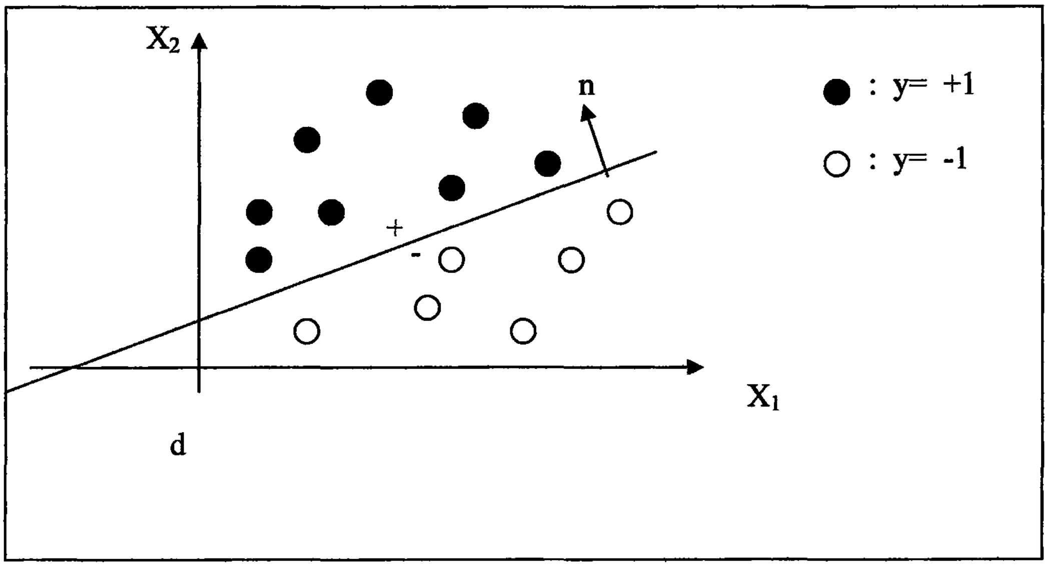


Figure 3.3 Perceptron Learning Rule Geometrical Visualization

CHAPTER FOUR
FUZZY LOGIC

4.1 Introduction to Fuzzy Logic

Fuzzy Logic is based on the way that human brain deals with inexact information as we meet in real world. Fuzzy logic can control or describe the system by using 'commonsense' rules that refer to indefinite quantities. These rules which the fuzzy logic is based on are of the form 'IF….. THEN ….' that map inputs to outputs, ie , one fuzzy set into another. Fuzzy logic utilizes linguistic variables rather than mathematical ones and applies these 'if … then…' rules for inference, simulating the human decision making.

The reasoning of conventional computers and logic rely on precise expressions which are reduced to strings of zeros and ones. The statements are evaluated as either true or false. However, real world is not constructed from sharp boundaries and crisp sets and human reasoning mostly depends on vague assertions and uncertainties. We define such an environment as 'fuzzy' and systems that suit this fuzziness can perform much better than conventional logic in this environment.

Fuzzy systems can be used when the mathematical model of the process does not exist, or too complex to be evaluated fast enough for real-time operation, and when high ambient noise levels must be dealt with. It can model non-linear functions of arbitrary complexity. We can use fuzzy systems to match any input-output data.

4.2 Fuzzy Logic And Fuzzy Inference

The underlying concept behind the fuzzy logic is 'everything is a matter of degree'. Fuzzy inference is the method of this interpretation of the values in the input vector and based on fuzzy rules, assigning values to the output vector.

In fact, fuzzy logic is a superset of classical two-valued logic. The curves defining how a point in input space (universe of discourse) is mapped to a membership value which are known as membership functions are in fact the truth value of a statement which the classical bivalent logic allows to be only 0 and 1. The logic operators such as AND, OR and NOT also find corresponding operators in fuzzy logic as min, max and additional complement.

Fuzzy Set-Theory, Fuzzy Relations, Fuzzy arithmetic and Fuzzy Logic are all generalizations of their corresponding fields in classical mathematics.

4.3 Fuzzy Set theory

The basic idea underlying is that, in a classical (non-fuzzy) set, the element of universe is either a member of the set or not. That is the membership of this element to the set is either 0 or 1. However, a fuzzy set allows this degree of membership to vary in the interval (0,1). Thus a classical crisp set has a unique membership function while a fuzzy set has infinite. A fuzzy set has no sharp boundaries that separates members and non-members. Transition between non-membership and full membership is gradual.

We can also define the complement, intersection, union etc. operations on fuzzy sets, and idempotence, distributivity, commutativity, associativity, absorption, law of zero and law of identity properties. We also have algebraic operations on fuzzy sets: negation, min and max operations.

4.4 Fuzzy Relations

The traditional 'crisp relation' is based on the fact that everything is either related or completely unrelated. But, 'fuzzy relations' allow relations with various degrees of interactions between elements. A fuzzy relation is a fuzzy set defined on Cartesian product of crisp sets $\{X_1, X_2, \ldots, X_n\}$ where $(x_1, x_2 \ldots, x_n)$ may have varying degrees of membership $\mu_R(x_1, x_2, \ldots x_n)$ within the relation.

Since the fuzzy relation from X to Y is a fuzzy set $X \times Y$, all the operations valid for fuzzy sets are also valid for fuzzy relations. But there are also some other operations valid only for relations.

Another important operation on fuzzy relations is the composition operations which can be applied to both relation-relation compositions and set-relation compositions.

CHAPTER FIVE
AN OVERVIEW OF CLUSTERING TECHNIQUES AND FUZZY C-MEANS CLUSTERING

5.1 Overview of Clustering Techniques

Clustering analysis is based on partitioning a collection of data points into a number of subgroups, where the objects inside a cluster show a certain degree of closeness or similarity. A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters.

Some of the requirements that a clustering method must satisfy are:

- Being able to deal with noisy data and outliers within the group
- Being independent of the order of input data
- Being able to handle multi-dimensional data
- Being able to deal with different types of attributes
- Being able to find clusters with different shapes (finding clusters different from spherical shape)

There are also a number of problems that clustering algorithms are subject to. Among them:

- Dealing with large number of dimensions and large number of data. Increased number of dimensions and data items means complexity and requires processing power and time;
- Results differ for each distance measure utilized. Selection of correct distance measure is important.

- Initial data and resultant clusters may need further interpretation. It's important to know whether clusters may be formed in the input data or not.

5.2 Classification of Clustering Techniques

Clustering techniques may be divided into two groups; Partitioning methods and hierarchical methods.

Hierarchical clustering proceeds successively by either merging smaller clusters into larger ones, or by splitting larger clusters. Hierarchical methods produce a set of nested clusters in which each pair of objects or clusters is progressively nested in a larger cluster until only one cluster remains.

The clustering methods differ in the rule by which it is decided which two small clusters are merged or which large cluster is split. The end result of the algorithm is a tree of clusters called a dendrogram, which shows how the clusters are related. By cutting the dendrogram at a desired level a clustering of the data items into disjoint groups is obtained.

Partitional clustering, on the other hand, attempts to directly decompose the data set into a set of disjoint clusters. Non-hierarchical methods divide a dataset of N objects into M clusters, with or without overlap. The criterion function that the clustering algorithm tries to minimize may emphasize the local structure of the data, as by assigning clusters to peaks in the probability density function, or the global structure. Typically the global criteria involve minimizing some measure of dissimilarity in the samples within each cluster, while maximizing the dissimilarity of different clusters.

Non-hierarchical methods are;

- Partitioning methods, classes are mutually exclusive
- Clumping method, overlap is allowed.

Hierarchical methods are;

- Divisive methods begin with all objects in a single cluster and at each of N-1 steps divide some clusters into two smaller clusters, until each object resides in its own cluster.
- Agglomerative methods: the hierarchy is build up in a series of N-1 agglomerations, or fusion, of pairs of objects, beginning with the un-clustered dataset.

5.3 Partitioning Methods

The partitioning methods generally result in a set of M clusters, each object belonging to one cluster. Each cluster may be represented by a centroid or a cluster representative. The center is usually the arithmetic mean of the attribute vectors for all objects within a cluster. If the number of the clusters is large, the centroids can be further clustered to produces hierarchy within a dataset.

Partitioning methods are divided according to the number of passes over the data.

- o Single pass (Basic partitioning methods)
- o Multiple passes (K –means, Very widely used)

5.3.1 Single Pass Algorithms

We can summarize generic single pass algorithms as follows:

- Make the first object the centroid for the first cluster.
- For the next object, calculate the similarity, S, with each existing cluster centroid, using some similarity coefficient.

- If the highest calculated S is greater than some specified threshold value, add the object to the corresponding cluster and re determine the centroid; otherwise, use the object to initiate a new cluster. If any objects remain to be clustered, return to step 2.

As its name implies, this method requires only one pass through the dataset; the time requirements are typically of order O(NlogN) for order O(logN) clusters. This makes it a very efficient clustering method. A disadvantage is that the resulting clusters are not independent of the order in which the documents are processed, with the first clusters formed usually being larger than those created later in the clustering run.

5.3.2 K-means Algorithm

In the k-means algorithm, given a set of numeric points in d dimensional space, and integer k Algorithm generates k (or fewer) clusters as follows:

- Assign all points to a cluster at random
- Compute centroid for each cluster
- Reassign each point to nearest centroid
- If centroids changed go back to stage 2

There are some weaknesses of k-means algorithm. First of all, although the k-means algorithm is simple and relatively fast to iterate it is a gradient descent method and therefore only capable of finding local energy minima. It will always converge on a low cost solution, but because the energy surface that it traverses is full of local minima, it will not necessarily find the global solution As such; it is extremely sensitive to the initial placement of exemplars. Exemplars are commonly placed randomly within the data space or randomly allocated from the data points themselves. It is therefore necessary to run the algorithm a number of times with different random initializations to try and find the best local minima possible.

Other weaknesses are; you must choose parameter *k* in advance, or try many values. Data must be numerical and must be compared via Euclidean distance (there is a variant called the *k*-medians algorithm to address these concerns). Also the algorithm works best on data which contains spherical clusters; clusters with other geometry may not be found. The algorithm is sensitive to outliers---points which do not belong in any cluster. These can distort the centroid positions and ruin the clustering.

<u>Initialization of k-means algorithm:</u> K-means results are highly dependent on the initialization procedure used. There are a number of different ways to initialize the algorithm:

1. Arbitrarily assign classes to objects. The straightforward way to do this is to assign the i^{th} object to the i modulo k^{th} class.
2. Initialize mean values with random objects from the data. To initialize k classes, choose k objects at random from the data. Make sure that the pair wise distance between the k distances is large enough.

<u>Termination of k-means algorithm:</u> Theoretically, k-means should terminate when no more objects are changing classes. There are proofs of termination for k-means. These rely on the fact that both steps of k-means (assign objects to nearest centers, move centers to cluster centroids) reduce variance. So eventually, there is no move to make that will continue to reduce the variance.

Running to completion (no objects changing classes) may require a large number of iterations. In software, we typically terminate when one of the following criteria is met:

1. Terminate when an error criterion is met
2. Terminate after fewer than a number of objects change classes
3. Terminate after a number of iterations.

<u>Dealing with dead classes:</u> Frequently, some classes become dead during the course of running k-means. A class is "dead" if no objects belong to it. There are a number of ways to deal with dead classes:

1. Mark a class as dead and never assign objects to it in the future.
2. Keep the mean of the class the same as it was before the iteration where no pixels were assigned to this class, in the hope that objects will be assigned to this class in the future.
3. Reassign the center of the class to a random value.

<u>How to measure quality of a classification:</u> One issue is how to measure the quality of the results provided by k-means classification. A good classification has:

Low within class variance -- compactness
High distance between class centers -- isolation

5.3.2.1 The algorithm

For a training set of length *M*, the segregation of input data into clusters is performed by minimization of the following cost function:

$$D = \frac{1}{M}\sum_{i=1}^{M} d_{\min}(X_i) = \frac{1}{M}\sum_{i=1}^{M} \min_{y\,j \in y} \left(d(x_i, y_j)\right) \qquad [5.1]$$

where *x* is the input data, *y* is the cluster centers, and *d* is the Euclidean distance:

$$d(x_i, y_j) = \left\| x_i, y_j \right\|^2 \qquad [5.2]$$

K means algorithm assigns each input to the nearest cluster in the means of Euclidean distance (Karayiannis, & Muralidhar, 1995).

According to the above formulation, the following membership function can be defined:

$$u_j(x_i) = \begin{cases} 1 & if\ d(x_i, y_j) = d_{\min}(x_i) \\ 0 & otherwise \end{cases} \qquad [5.3]$$

Cluster centers are evaluated iteratively as the mean of the input vectors assigned to each cluster.

$$y_j = \frac{\sum_{i=1}^{M} u_j(x_j)x_i}{\sum_{i=1}^{M} u_j(x_j)} \quad and \quad j = 1,2,\ldots\ldots,k \qquad [5.4]$$

5.3.3 Sequential K means

A way to modify the k-means procedure is to update the means one example at a time, rather than all at once. This is particularly attractive when we acquire the examples over a period of time, and we want to start clustering before we have seen all of the examples. The algorithm is similar to the regular k means algorithm. The algorithm is:

Make initial guesses for the means $m_1, m_2, \ldots, m_k$
Set the counts $n_1, n_2, \ldots, n_k$ to zero
    Until interrupted
        Acquire the next example, x
            If m_i is closest to x
                Increment n_i
                Replace m_i by $m_i + (1/n_i)*(x - m_i)$
            end_if
        end until

5.3.4 K - Medoids clustering method

In K-medoids clustering method, the most centrally located objects in a cluster are chosen as the medoid. The algorithm is:

Arbitrarily choose k objects as the initial medoids
    Until no change, do
        (Re)assign each object to the cluster to which the nearest medoid
        Randomly select a non-medoid object o', compute the total cost, S, of swapping medoid o with o'
            If $S < 0$ then swap o with o' to form the new set of k medoids

$$S = \sum_{clusters} \sum_{points} d(p,o)$$

            end_if
    end_until

5.4 Fuzzy C-Means Clustering Algorithm

Clustering analysis is based on partitioning a collection of data points into a number of subgroups, where the objects inside a cluster (a subgroup) show a certain degree of closeness or similarity. Hard clustering (ex, standard k-means) assigns each data point (feature vector) to one and only one of the clusters, with a degreee of membership equal to one, assuming well defined boudaries between the clusters. The clusters produced by the hard clustering procedures are sometimes called "hard" or "crisp" clusters.

This model does not reflect the description of real data, where boundaries between subgroups might be fuzzy. It's in contrast to "soft" or "fuzzy" clusters, in which a feature vector $\boldsymbol{x}$ can have a degree of membership in each cluster. The fuzzy k-means procedure of Dunn and Bezdek allows each feature vector $\boldsymbol{x}$ to have a degree of membership in a cluster.

Fuzzy set theory is a method of representing vagueness in every day life. Fuzzy clustering algorithms consider each cluster as a fuzzy set, while a membership function measures the possibility that each training vector belongs to a cluster. As a result, each training vector may be assigned to multiple clusters with some degree of certainty measured by the membership function. Thus, the partition of the training set is based upon soft decisions (Karayiannis, & Muralidhar, 1995).

Fuzzy c-means clustering algorithm is an unsupervised algorithm. Unsupervised clustering algorithms are advantageous especially when

- We don't have labeled data
- We don't have information about number of classes, a priori probability etc.

5.4.1 Characteristics of Fuzzy c-means Algorithm

The fuzzy k-means algorithm is simply an iterative operation for finding memberships of all the objects in the feature space that optimize the objective function. The fuzzy k-means algorithm follows the same steps as the conventional version of the clustering algorithm. However, when the objects are assigned to clusters they are also assigned a membership function that describes the degree of membership to that cluster. As with fuzzy sets, each object can be a member of more than one cluster with membership functions for each. Each object-to-centroid (mean object in cluster) difference is weighted by a function of that object's membership value in that cluster. The membership functions are then iteratively altered to minimize the sum of weighted differences.

The objective function for Fuzzy c-means clustering algorithm is (Bezdek, Ehrlich, & Full, 1984):

$$J(U,D) = \sum_{i=1}^{n} \sum_{j=1}^{k} u_{ij}^{m} d_{ij}^{2} \qquad [5.5]$$

where U is the membership function:

$$u_j(x_i) = \frac{1}{\sum_{l=1}^{k} \left(\frac{d(x_i, y_j)}{d(x_i, y_l)} \right)^{\frac{1}{m-1}}} \qquad [5.6]$$

or similarly expressed as:

$$u_j(X_i) = \frac{(1/(d(x_i y_j)^2)^{1/m-1}}{\sum_{j=1}^{c} (1/d(x_i y_j)^2)^{1/m-1}} \qquad [5.7]$$

Y is the cluster centers updated at each iteration according to the following formula:

$$y_j = \frac{\sum_{i=1}^{M} u_j(x_j)^m x_i}{\sum_{i=1}^{M} u_j(x_j)^m} \quad and \quad j = 1,2,\ldots\ldots,k \qquad [5.8]$$

The degree of fuzziness in the above formulae is determined by the parameter: m, which is greater than one (Bezdek, Ehrlich, & Full, 1984). As this parameter approaches one, fuzziness degrades and algorithm approaches to standard k-means algorithm (Bowden, 2000).

The membership function is first weighted, which is present to reduce noise in the data. This exponent weight reduces the influence of small membership values. The effect of noise is also reduced in calculation of the cluster centroid. Both the membership functions and fuzziness index are included in the equation to reduce the effect of outliers on the centroid. This algorithm is a generalization of the conventional algorithm and as a result is more flexible and capable of handling practical applications of clustering. Fuzzy sets have been found to be an important tool in pattern recognition applications. This includes fields such as speech recognition, image processing, character recognition and medical applications. This is because fuzzy set theory is well suited to modeling human cognitive processes in particular those related to recognition. In addition to this, fuzzy set theory has provided many new algorithms for the classification of objects (Pedrycz, 1990).

5.4.2 The algorithm

Detailed Fuzzy c-means algorithm is as follows:

Step 1:

Given a data set,

- Select the number of clusters
- Initialize the cluster centroids, w_j ($2 < j < c$)
- Initialize the fuzzification parameter, m ($1 < m < \infty$)
- Initialize the termination tolerance, $e > 0$
- Initialize the fuzzy membership matrix U to $U(0)$, $U(0)$ e U

Step 2:

- Calculate the Euclidean distance, $d_{i,j}$ between each training sample x_i and the class centroid w_j

- Calculate the membership matrix U, using the equation 5.7,

$$u_j(X_i) = \frac{(1/(d(x_i y_j)^2)^{1/m-1}}{\sum_{j=1}^{c}(1/d(x_i y_j)^2)^{1/m-1}}$$

Step 3:

- Update the class centroids according to equation 5.8,

$$y_j = \frac{\sum_{i=1}^{M} u_j(x_j)^m x_i}{\sum_{i=1}^{M} u_j(x_j)^m} \quad and \quad j = 1,2,\ldots\ldots,k$$

Step 4:

- Compute $\Delta = \max(|U(t+1) - U(t)|)$.
  - If $\Delta > e$, then go to Step 2;
  - Otherwise go to Step 5.

Step 5:

- Find the results for the final class centroids.

Fuzzy c-means algorithm has the advantage that it more naturally handles situations in which subclasses are formed by mixing or interpolating between extreme examples, so that it makes more sense to say that x is 40% in Cluster 1 and 60% in Cluster 2, rather than having to assign x completely to one cluster or the other.

5.4.3 Parameters In Fuzzy C-means Clustering Algorithm

Some parameters are involved during the processing of the algorithm. These parameters need to be adjusted before the algorithm is started.

Fuzziness parameter: This parameter determines fuzziness of the resulting partition. It's value must be equal to or greater than one. If it's equal to one, algorithm is no longer fuzzy. This parameter also reduces the sensitivity to noise.

Generally the default value of this parameter is 2.

Error criterion: Error criterion determines the point algorithm will stop when the error function reaches. Usually, it's selected as 0.001. As the value gets smaller, sensitivity increases but this leads to more computational times as normal. Choosing the error criterion as 0.01 drastically reduces the computing times

Iteration Number: This is the number of maximum iterations algorithm will run if error criterion can not be met.

5.5 Distance measures

An important component of a clustering algorithm is the distance measure between data points. If the components of the data instance vectors are all in the same physical units then it is possible that the simple Euclidean distance metric is sufficient to successfully group similar data instances. However, even in this case the Euclidean distance can sometimes be misleading. The problem arises from the mathematical formula used to combine the distances between the single components of the data feature vectors into a unique distance measure that can be used for clustering purposes; different formulas leads to different clustering. Again, domain knowledge must be used to guide the formulation of a suitable distance measure for each particular application. Below are two examples of different distance measures:

Euclidean Distance: The square root of the component wise square of the difference between the points.

$$d(x,y) = \sqrt{\sum_{i=1}^{p} |x-y|^2} \qquad [5.9]$$

Manhattan Distance: The absolute value of the component wise difference between the points. This is the simplest distance to calculate and may be more robust to outliers.

$$d(x,y) = \sum_{i=1}^{p} |x-y| \qquad [5.10]$$

In this study Euclidean distance is used.

CHAPTER SIX
THE APPLICATION: EEG SPIKE DETECTION WITH FUZZY C-MEANS CLUSTERING

6.1 Introduction

In this study, electroencephalogram (EEG) recordings are examined to detect epileptiform spikes. As mentioned in Chapter 2, an epileptiform spike generally has a duration between 20 and 70ms, and the apex of the spike is sharp (Acır, 2004). Spike may be seen alone but it may also be followed by a slow wave. In this case the signal is called a 'spike and slow wave complex' (SSWC) and duration is between 150 and 350ms. Automatic detection of spikes is extremely important for diagnosis of epilepsy and is the aim of this study.

As mentioned in Chapter 2, data may be presented to the EEG spike detector in two forms, raw EEG data or extracted features of the EEG data. Presenting raw EEG data to the system is not much practical since processing of these data requires relatively long time and big processing power. However, this method avoids any data loss that parameterization techniques will introduce.

Parameterization of EEG data means extracting specific features from the spikes. In this method, success of the spike detection relies on the correct selection of features. These features may include frequency domain parameters like linear prediction parameters or total power, or time domain parameters like amplitude, duration and slope.

Generally EEG recordings are first filtered against artifacts. Inter-ictal spiking is generally characterized as 50-270 ms duration spikes. So 50Hz and higher frequencies may be filtered by a low pass filter with 50 Hz cut off frequency. Proper windowing of these signals eliminate excessive data and allows us to deal with abnormalities in the recording.

In this study, epileptiform spike detection system consists of two stages. The first stage is data pre-processing and pre-classification part. Signal processing techniques are utilized for extracting peaks from EEG recordings and extracting time-domain features from the peaks. For pre-classification part, a simple neural network model consisting of two perceptrons is employed.

Second stage is post classification with fuzzy c-means clustering algorithm.

All the software is implemented using Matlab® 6.5 (r13) and related neural network and fuzzy c-means toolboxes. Software is included in Appendix A.

6.2 Data Acquisition

Data used in this study consists of 7 EEG records obtained as 19 channels with 256 Hz sampling frequency. These EEG recordings are labeled for epileptiform spikes by electroencephalographers. Only a single channel with epileptiform spikes (ES) are used from each record in this study. There are a total of 55 epileptiform spikes in the input data.

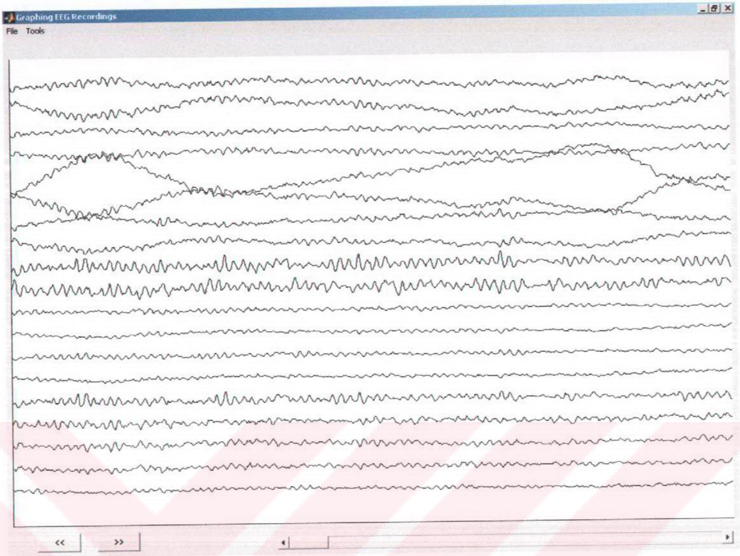


Figure 6.1 Example of input EEG record

Among these 7 recordings, 4 are used for neural network training and as reference data for fuzzy c-means clustering. These data have a total of 40 ES. Other 3 recordings are used for testing and have 15 ES.

Another set of input data consists of 6 extracted features of 100 spikes. These spikes are the output of the pre-classification part in Dr. Nurettin Acır's Ph.D. Thesis and included in this study for comparison purposes (Acır, 2004). This data is classified as either non-definite epileptiform spike or non-definite non-epileptiform spike by the pre classification stage consisting of two perceptrons. These data are applied as input to post classifier in the first method explained in section 6.5 and fuzzy c-means classification performance is examined.

The EEG data used in this study is provided by Dr. Nurettin Acır. Each EEG record is provided as a separate text file on a compact disc.

6.3 Data pre-processing

As Matlab® is used for programming, raw EEG data in the text files are converted to vector form:

$[p_1^{\,1}\ \ p_1^{\,2}\ \ldots\ldots p_1^{\,n}\ ]$ ---- > Signal 1

$[p_2^{\,1}\ \ p_2^{\,2}\ \ldots\ldots p_2^{\,n}\ ]$ ---- > Signal 2

where n is the signal length.

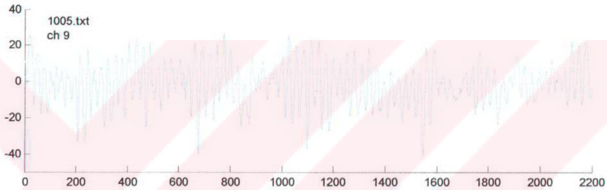


Figure 6.2 Sample EEG data (single channel)

The part of the EEG record illustrated above consists of 2200 data points; i.e., n=2200. The length of the whole record is 10240. A smaller part of the record with 100 data points demonstrates the EEG signal construction from the measured data.

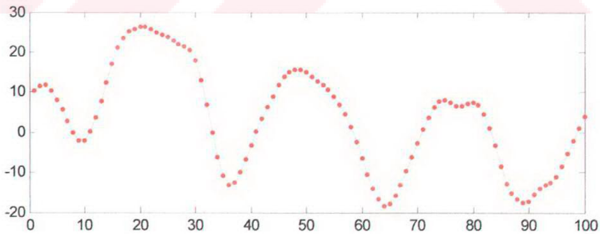


Figure 6.3 EEG signal with construction from the measured data points

First step of pre-processing stage is the normalization of data. We find the average of each individual signal and subtract it from the signal itself.

Second step is extracting peaks. As epileptiform waves generally reveal themselves in form of transient spikes with a pointed peak, we need to determine the peaks first, then decide whether it's epileptiform or not.

For peak detection, we calculate the amplitude difference between each data point. This gives us the slope between the data points. If the slope is positive, signal is increasing. If it's negative, signal is decreasing. Whenever slope changes from positive to negative or vice versa, this means there is a change in signal direction, that is, there exists a peak. We can formulate this as below:

If P[n] – P[n-1] >0 AND P[n+1] – P[n] <0, this is a peak where signal slope changes from rising to falling. (This is Positive Peak)

If P[n] – P[n-1] <0 AND P[n+1] – P[n] >0, this is a peak where signal changes from falling to rising. (Negative Peak)

If P[n] – P[n-1] >0 AND P[n+1] – P[n] >0

or P[n] – P[n-1] <0 AND P[n+1] – P[n] <0, this point is not a peak.

This calculation is an approximation of finding the time derivative of an analog signal. Where the derivative is positive, signal is increasing, where derivative is negative signal is decreasing. If the derivative changes from positive to negative or vice versa, this point is a positive or negative peak in the analog signal.

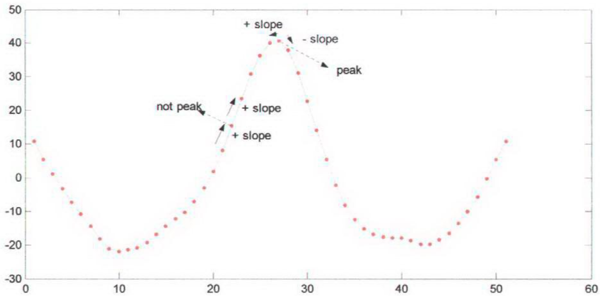


Figure 6.4 Extraction of peaks

In order to eliminate artifacts from the signal and ignore small changes with higher frequency we use a simple algorithm:

If the length of signal between two peaks is shorter than the length of signals of previous and next peaks, and if duration of this signal part is shorter than 20ms and amplitude of this part is smaller than 2μV, we eliminate this peak. This procedure is demonstrated in the following figure:

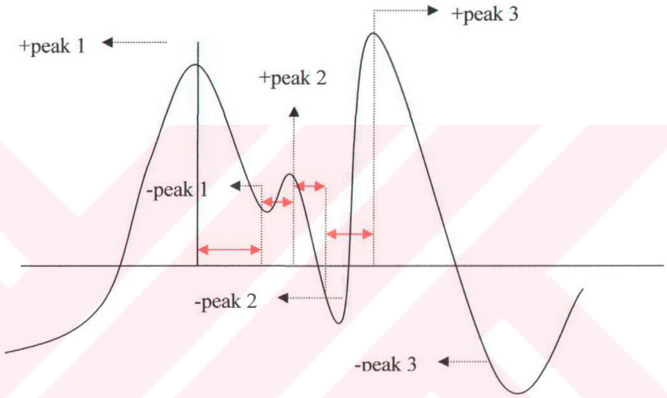


Figure 6.5 Visualization of peak elimination

As illustrated in Figure 6.5, positive peak number two has a shorter duration then the previous and next peaks. If this duration is less than 20ms and if it's amplitude is smaller than 2μV, this peak will be removed (Acır, 2004).

This process also removes spikes with a frequency of 50 Hz and higher. This filtering eliminates mains electricity interference and sharp, short spikes that result from patient movement.

As the peaks are detected, now we can remove each individual spike from the record with proper windowing or we can extract time-domain parameters of each spike. For applying raw data as input, a window with 40 data points length is used

(Acır, 2004). This corresponds to approximately 150 ms of the EEG signal and is an average value for the spike duration. However, for applying extracted features of each spike as input, no windowing will be applied since peaks will not be removed from the record.

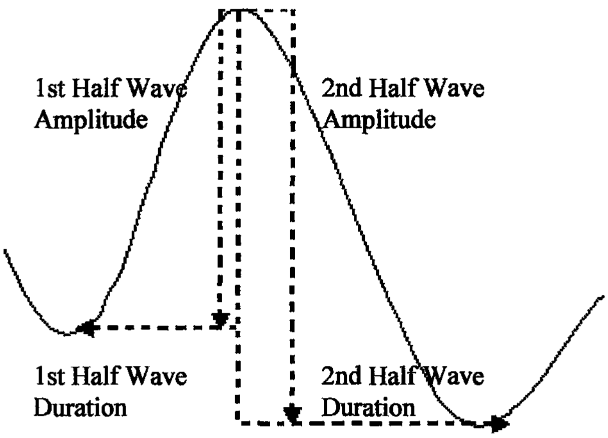


Figure 6.6 Features extracted from spikes

The extracted features in this thesis are time-domain parameters of the spikes. We consider the spikes in two half waves. The duration of the first and second half waves are found by extracting the duration between the positive peak of the spike and the previous and next peaks. Again the amplitude of the spike is found by calculating the difference between the amplitudes of peak of spike and previous and next peaks.

In the software, these features are symbolized as:

1st half wave duration: *dp* 2nd half wave duration: *dn*

1st half wave amplitude: *mp* 2nd half wave amplitude: *mn*

The first half wave slope and second half way slope are evaluated by:

1st half wave slope: sp = mp/dp 2nd half wave slope: sn = mn/dn

These features are stored in matrix form:

$$
\begin{array}{l}
\begin{pmatrix}
dp^1 & dn^1 & mp^1 & mn^1 \\
dp^2 & dn^2 & mp^2 & mn^2 \\
\cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots \\
dp^m & dn^m & mp^m & mn^m
\end{pmatrix}_{mx4}
\end{array}
\begin{array}{l}
\text{----} > \text{Spike } 1 \\
\text{----} > \text{Spike } 2 \\
\\
\\
\text{----} > \text{Spike } m
\end{array}
$$

In the calculation of extracted features, four of the above properties (*mp, mn, dp, dn*) are used. For comparison purposes, data out of a pre classification stage from Dr. Nurettin Acır's Ph. D. thesis is utilized in one of the approaches and slope is also evaluated in this data in addition to duration and amplitude.

6.4 Pre-classification:

In the first classification stage, the aim is to reduce data by eliminating definite epileptiform spikes (ES) and definite non-epileptiform spikes (non – ES). A simple artificial neural network structure with two discrete perceptrons is used for this classification. Network is single layer and has four inputs and a bias input.

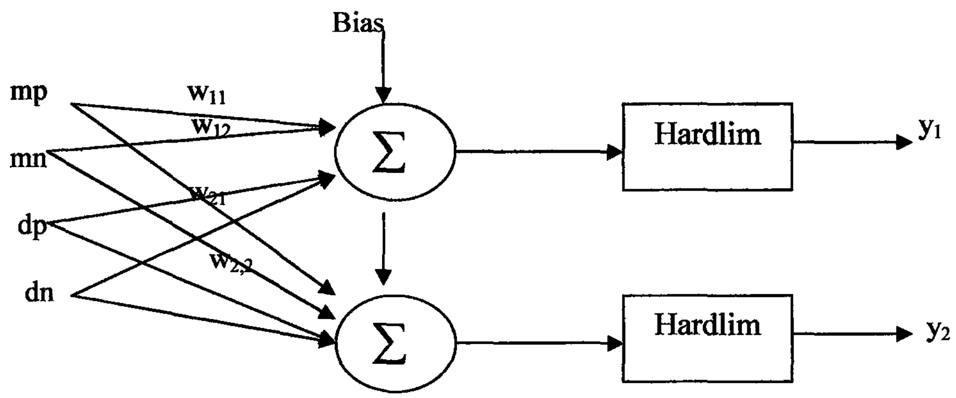


Figure 6.7 Neural network structure used in pre-classification stage

'Hardlim' activation function is used in the network. The output of the perceptrons is calculated using the following formula where y is the output of perceptrons, w is the weight vector, x is the input vector and τ is the threshold (bias).

$$y^S = \text{hardlim}(\mathbf{w}^{\mathbf{T}}\mathbf{x}^{\mathbf{s}} - \tau) \qquad \text{hardlim}(u) = \begin{cases} 1 & u \geq 0 \\ 0 & u < 0 \end{cases} \qquad [6.1]$$

In the software implementation of this neural network;

- Two node perceptron network implemented Matlab®
- Two node perceptron network implemented with Matlab® neural network toolbox for comparison.

As perceptron learning rule is supervised, we need to define the desired outputs for the training of the perceptrons. Desired output for one of the perceptrons is +1 for definite ES's and 0 for the rest. For the second perceptron, desired output is +1 for definite non-ES's and 0 for the rest of inputs.

Table 6.1 Perceptron output mapping

	+1	0
P1	ES	Non definite ES/non-Es
P2	Non ES	Non definite ES/non-ES

A data set of 40 Epileptiform Spikes (ES) is prepared for training. These spikes are integrated with raw EEG data with 102 non-ES and 3 ES. So, a total of 43 ES and 102 non ES data are used for the training.

After training, the network is tested with a test data with a total of 774 spikes and 15 ES. At the end of the testing, input spikes are separated into three groups. The first group that first perceptron outputs +1 is ES. The second group that the second perceptron outputs +1 is non-Es. The third group of spikes consists of non-definite ES's and non-definite non-ES's and both perceptrons output 0 for these spikes. This group forms the input of the post classifier (Acır, 2004). This group has 41 spikes with 2 ES.

6.5 Fuzzy c-means clustering

The aim of the post-classifier is to separate the peaks in the third group. To realize this clustering, two different software implementation of fuzzy c-means clustering algorithm is applied. One of the programs is written using Matlab® fuzzy c-means toolbox. The other program, 'Fuzme' is a generic fuzzy c-means classifier software downloaded from internet. This second software is utilized for comparison purposes.

In the post classification stage different approaches are used to investigate the performance of this second part and the overall system. Three data sets are used as input to realize these approaches.

6.5.1 Input Data

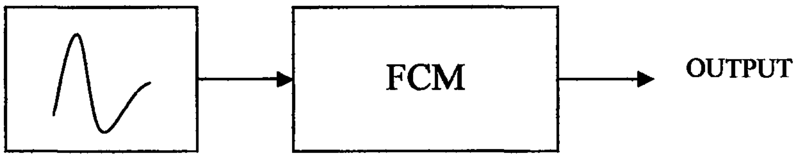
Data set 1: Data obtained from 7 EEG records with 19 channels is utilized in the rest of the application. Following approaches are tested with this data set:

Data set 2: A data set of six extracted features from 100 spikes mentioned in section 6.2 is provided as input to the system. These spikes are the output of the pre-classification stage utilized in the Ph. D. thesis of Dr. Nurettin Acır and found to be non-trivial spike or non-trivial non-spike.

Data set 3: Data used for training of the perceptrons in the first stage are used as reference input in 'approach 4'. This data consists of 43 ES.

6.5.2 Employed Methods

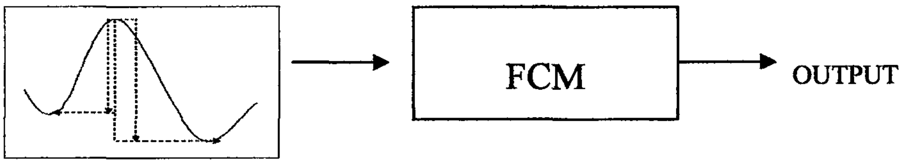
Approach 1 (Data set 1): Raw data extracted from the EEG records are used as input. Individual spikes are extracted from the records with a window length of 40 data points. A total of 75 spikes are captured from the EEG's with 38 definite ES. Then these data are applied to fuzzy c-means classifier without a pre-classification.



RAW DATA

Figure 6.8 Block diagram of 1st approach

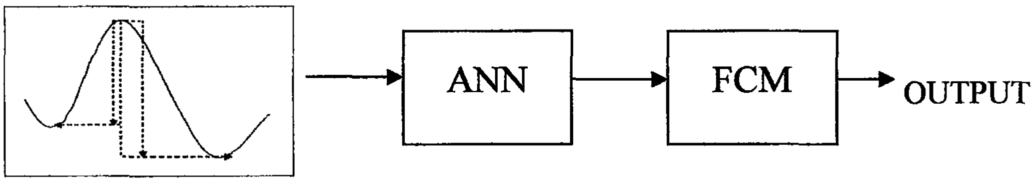
Approach 2(Data set 1): Spikes captured in the first approach are subjected to feature extraction. Four time domain parameters are extracted for each spike and this data set is applied to fuzzy c-means classifier without pre-classification.



PARAMETERIZED DATA

Figure 6.9 Block diagram of 2nd approach

Approach 3 (Data set 1): A set of 166 individual spikes are captured from the EEG records. Four time domain parameters are extracted from each spike and this data is used for training and testing of the two-stage classifier. The third group of spikes classified as non-definite epileptiform spikes or non-definite non-epileptiform activity in the pre-classification stage are then fed to the fuzzy c-means classifier. This data was saved in a text file after the extraction and re-loaded during the initialization part of the second stage.



PARAMETERIZED DATA

Figure 6.10 Block diagram of 3rd approach

Approach 4 (Data set 1 & Data set 3): In addition to data used in approach 3, a data set consisting of 125 spikes and 43 ES is used as reference input to the fuzzy c-means classifier since data set 1 has only 2 ES.

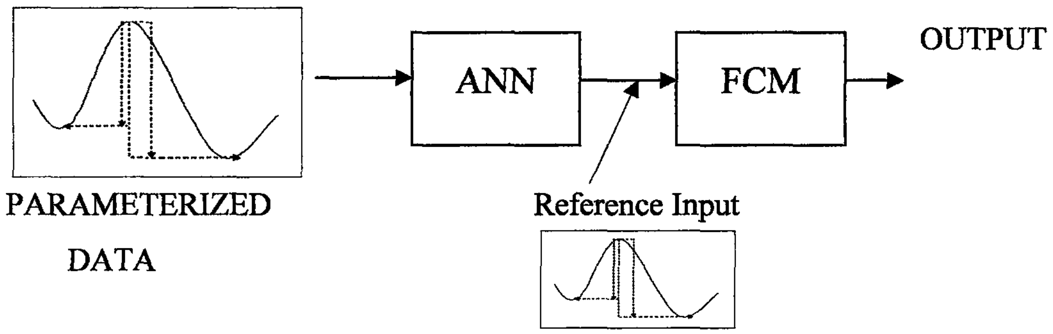


Figure 6.11 Block diagram of 4th approach

Approach 5 (Data set 2): Data is applied to the fuzzy c-means classifier without pre-classification.

6.5.3 Fuzzy c-means clustering programming

For the fuzzy c-means clustering software,
[CENTER, U, OBJ_FCN] = FCM (DATA, CLUSTER_N, OPTIONS)
command from Matlab® fuzzy c-means toolbox is used.

This command accepts the data to be clustered and number of classes as input arguments. OPTIONS are a set of optional inputs and determine:

CLUSTER_N: Number of classes.
OPTIONS (1): exponent for the partition matrix U (default: 2.0)
OPTIONS (2): maximum number of iterations (default: 100)
OPTIONS (3): minimum amount of improvement (default: 1e-5)
OPTIONS (4): information display during iteration (default: 1)

Initial centroids for the clusters are selected randomly. Euclidian distance is employed as the similarity measure.

6.6 Results

Results of this study are evaluated by the following parameters (Pang, Upton, Shine, & Kamath, 2003):

True Positives – TP: Number of epileptiform spikes detected

True Negatives – TN: Number of non-epileptiform activity detected

False Positives – FP: Number of activity incorrectly labeled as epileptiform

False Negatives – FN: Number of spikes incorrectly labeled as non-epileptiform

Sensitivity (SEN): Ability to detect ES

Specifity (SPE): Ability to specify normal activities

Selectivity (SEL): Ability to reject false ES detection

Average detection rate (ADR): Average of sensitivity and specifity

Approach 1 (Data set 1):

Feeding raw input data to the fuzzy c-means clustering software is tried in this method. The data set has 75 samples of spikes with 38 true ES.

Table 6.2 Results of first approach

ES labeled By sw	TP	TN	FP	FN	SEN (%)	SPE (%)	SEL (%)	ADR (%)
44	28	21	16	10	73,7	56,7	63,6	65,2

Approach 2 (Data set 1):

Also feeding input data to the fuzzy c-means classifier without pre-classification stage is tested. Input set is formed from the feature vector of 75 spikes with 38 ES.

Table 6.3 Results of second approach

ES labeled By sw	TP	TN	FP	FN	SEN (%)	SPE (%)	SEL (%)	ADR (%)
46	32	23	14	6	84,2	62,1	69,5	76,85

Approach 3 (Data set 1):

Input data used in the third approach is the extracted feature set of the spikes classified as the third group in the pre-classification stage mentioned in 6.4. This data consists of 41 spikes with 2 real ES.

The output of fuzzy c-means classification algorithm for each spike is a vector (U) that determines the grades the spike belongs to the clusters. For two clusters (N=2), output is a two row vector, first row stating the ratio the spike belong to the first cluster and second raw stating the ratio the spike belongs to the second cluster. That's why the sum of the rows is 1. These values are called membership values.

After the classification is completed and the membership values (U) are examined, the outputs are filtered by software so that only spikes having 0,8 or higher membership to a cluster is counted to be in that cluster. The aim is only to get the spikes belonging to a cluster with a high membership value. After this filtering, 10 spikes are left in one of the clusters and 2 spikes left in the second cluster. The two ES are in the cluster with 10 spikes and the spikes have the following membership values:

Table 6.4 Membership values of some of the resultant ES in the third approach (N=2)

	NonES	NonES	NonES	NonES	NonES	**ES 1**	**ES 2**	NonES
C1	0,1167	0,1330	0,1036	0,1420	0,1625	**0,1960**	**0,1549**	0,1206
C2	0,8833	0,8670	0,8964	0,8580	0,8375	**0,8040**	**0,8451**	0,8794

Table 6.5 Results of third approach (N=2)

ES labeled by sw	TP	TN	FP	FN	SEN (%)	SPE (%)	SEL (%)	ADR (%)
10	2	3	8	0	93,3	94,1	24,5	93,7

FCM Parameters set in this test:

Fuzzy exponent (Phi) = 3

Maximum iterations = 50000

Convergence criterion =0,0001

Scatter around initial membership =0,2

Number of trial to choose an optimal solution =10000

Since selectivity is very low, another trial is made by increasing the final number of clusters to see if the ES are still classified in a single cluster. The threshold value for a spike to belong to a class is decreased to 0,55.

Table 6.6 Results Membership values of resultant ES in the third approach (N=4)

			ES 1	**ES 2**	
C1	0,2693	0,1024	**0,2384**	**0,1844**	0,2471
C2	0,0403	0,0193	**0,0535**	**0,0359**	0,0389
C3	0,5816	0,8291	**0,5895**	**0,6855**	0,6126
C4	0,1087	0,0492	**0,1186**	**0,0941**	0,1014

Table 6.7 Results of third approach (N=4)

ES labeled by sw	TP	TN	FP	FN	SEN (%)	SPE (%)	SEL (%)	Select. of FCM (%)	ADR (%)
5	2	36	3	0	93,3	94,8	26,4	40	94,05

FCM Parameters set in this test:

Fuzzy exponent (Phi) = 3

Maximum iterations = 50000

Convergence criterion =0.0001

Scatter around initial membership =0,2

Number of trial to choose an optimal solution =10000

Approach 4 (Data set 1 & Data set 3):

Since number of epileptiform spikes is too low, a reference input with 125 spikes and 43 ES is also presented to the system to see the improvement. Threshold of membership value for a spike to belong to a cluster is set to 0,55. Total number of spikes in the data set is 166. Number of true ES is 45.

Table 6.8 Results of fourth approach

ES labeled by sw	TP	TN	FP	FN	SEN (%)	SPE (%)	SEL (%)	Select. of FCM (%)	ADR (%)
59	43	98	17	2	94,8	91,5	51,4	71,1	93,15

Approach 5 (Data set 2):

A set of 100 samples of spikes is obtained with 6 true ES. 6 time domain features extracted from the spikes are used as input to the system. After the fuzzy c-means algorithm is run, the following results are obtained. (ES: Definite epileptiform spikes)

Table 6.9 Results of fifth approach

ES labeled by sw	TP	TN	FP	FN	SEN (%)	SPE (%)	SEL (%)	ADR (%)
17	6	83	11	0	100	88,3	35,3	94,15

In order to compare the performance of this program, the second program downloaded form internet is employed. This program is FUZME Fuzzy c-means clustering software provided by Australian Centre for Precision Agriculture (ACPA) and the University of Sydney. (2003)

This software is fed by the same input vector consisting of six features and the results are exactly the same as the previous software.

6.6.1 Comparison Of Results:

The results for all these approaches are summarized in the following table:

Table 6.10 Comparison of results

	Appr 1	Appr 2	Appr 3 (k=2)	Appr 3 (k=4)	Appr 4	Appr 5
Sensitivity	73,7	84,2	93,3	93,3	94,8	100
Specifity	56,7	62,1	94,1	94,8	91,5	88,3
Selectivity	63,6	69,5	24,5	26,4	51,4	35,3
ADR	65,2	76,85	93,7	94,05	93,15	94,15

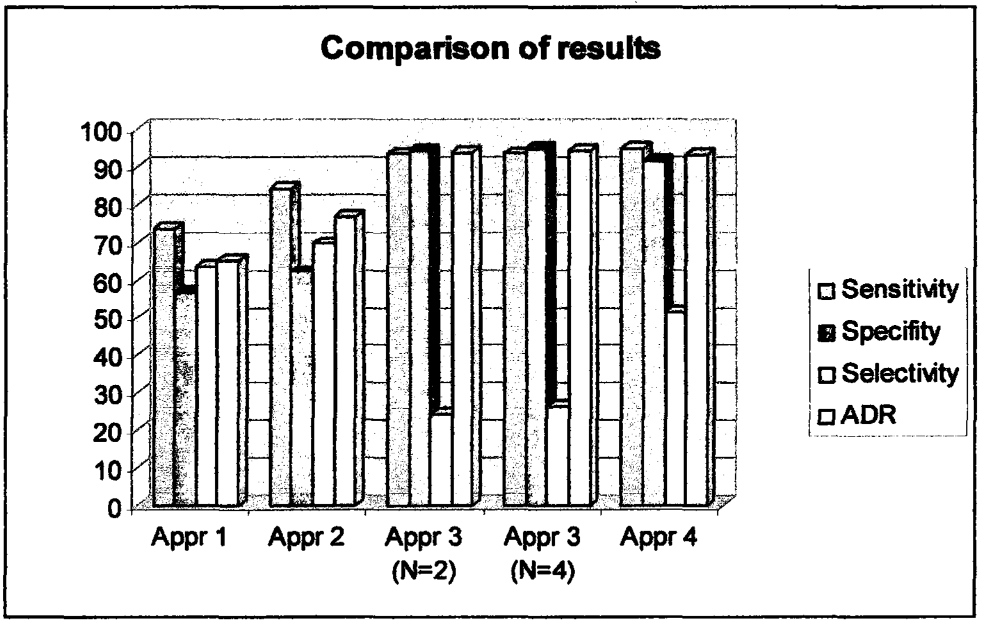


Figure 6.12 Graphical demonstration of results

6.6.2 Comparison of Results with Previous Researches:

There are many different techniques and methods used for the epileptiform spike detection purposes. For example, Tarassenko used extracted features of EEG data as input. These features include slope, sharpness, mobility and complexity (Pang, Upton, Shine, & Kamath, 2003).Webber, Litt, Wilson, & Lesser (1994) utilized artificial neural networks and mimetic methods are used for classification. Özdamar, & Kalayci (1995) also used neural network based systems for classification purposes and reported very satisfactory results. Dingle, Jones, Carroll, & Fright (1993) and James, Jones, Bones, & Carroll (1999) employed multi-stage methods.

In the following table results of these studies are compared with the results of the method proposed in this thesis.

Table 6.11 Comparison of various methods used in EEG spike detection (Pang, Upton, Shine, & Kamath, 2003)

Researches	Sensitivity (%)	Specifity (%)	Selectivity (%)	ADR (%)
Weber	86,61	95,34	86,32	90,98
Özdamar	84,38	94,48	82,84	89,43
Tarassenko	84,15	95,05	84,27	89,59
Dingle	100	-	58	-
James	82	-	55	-
Acır	91,4	-	88,6	-
FCM				
Appr 1	73,7	56,7	63,6	65,2
Appr 2	84,2	62,1	69,5	76,85
Appr 3 (k=2)	93,3	94,1	24,5	93,7
Appr 3 (k=4)	93,3	94,8	26,4	94,05
Appr 4	94,8	91,5	51,4	93,15
Appr 5	100	88,3	35,3	94,15

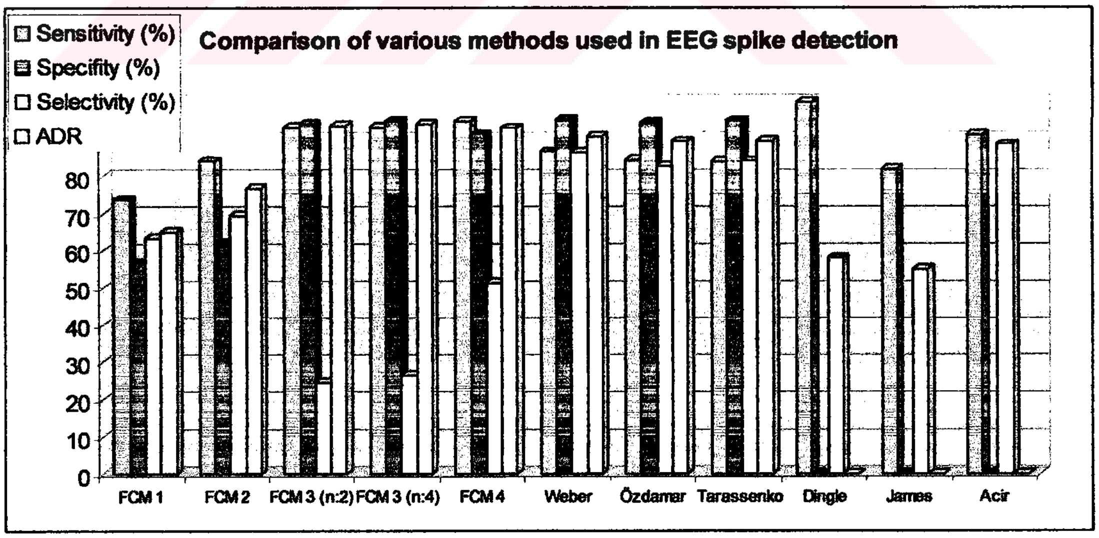


Figure 6.13 Graphical demonstration of comparison of results with various methods

CHAPTER SEVEN
CONCLUSION

7.1 Comments

This thesis is based on fuzzy c-means clustering algorithm and artificial neural networks for detection of epileptiform waves in electroencephalogram records. Epileptiform waves generally reveal themselves in form of spikes and they have duration between 20 and 70ms, and the apex of the spike is sharp. These characteristics of epileptiform activities in EEG are utilized in this study for detection purposes.

The spike detection method used in this study has two stages; pre-processing and pre-classification of input data and post classification. Preprocessing consists of determination of peaks, getting peaks from the records with a proper window size or extracting time domain features of these spike-like waves. Pre-classification is formed by a simple artificial neural network structure with elementary nodes known as perceptrons. Input data applied to pre-classification part is divided into three groups; definite epileptiform spikes, definite non-epileptiform spikes and non-definite epileptiform spikes and non-definite non-epileptiform spikes.

A clustering technique known as fuzzy c-means classification is employed for the post classification of third group of data mentioned above which consists of non-definite epileptiform spikes and non-definite non-epileptiform spikes. However, other alternative methods are examined to investigate the system performance like usage of fuzzy c-means (FCM) classification without pre-classifier stage or presenting reference input data to FCM for better clustering.

The performance of the system is evaluated by sensitivity, selectivity, specifity and average detection rate parameters. (Pang, Upton, Shine, & Kamath, 2003)

Main characteristics of this study include:

- Two stage algorithm with data pre-processing and pre-classification
- Usage of both parameterized and raw input data
- Single channel spike detection
- Outputs of FCM are not discrete, i.e., each spike has a membership value to a cluster implying a ratio of being ES or non-ES.
- Supervised method for pre-classification and unsupervised method for post-classification
- Offline (batch) processing

In the first and second approach utilized in this study, pre-classification stage is not used. First approach uses raw data and second approach uses parameterized data. When the results are compared, all performance parameters are higher when parameterized data is used. Input data dimension for raw data is 40 since a window size of 40 data points is used for each spike whereas input data dimension for parameterized data is 4 while number of input samples remains the same. Number of input samples might be increased for an improvement in the results.

For the third approach, the input data is first processed to extract spikes and some time-domain features of these spikes. These spikes are pre-classified by the first stage in the system consisting of two perceptrons. Each perceptron divides the input data into two groups; one of the perceptrons output +1 for ES and 0 for other activities. The other perceptron outputs +1 for non-ES and 0 for other activities. So the spikes that both perceptrons output 0 are non-definite activity, either ES or non-ES. These spikes are then fed to the fuzzy c-means classification process.

The results of this method are quite satisfactory in terms of sensitivity and specifity but very poor in terms of selectivity. This is because of the high number of false positives detected by FCM algorithm.

FCM classifies inputs into groups with respect to the Euclidian distances between samples. That's why more characteristic inputs mean better classification. However, in this application input data of FCM is formed by non-definite signal shapes. The algorithm can recognize the epileptiform spikes that have characteristic features but does not have enough samples to separate them from other inputs.

This idea is also the foundation of the fourth approach. A set of reference data including many epileptiform spikes is presented to FCM together with the output data of pre-classification. The aim is to provide characteristic spike features to FCM for better classification. The results show the improvement in selectivity with a relatively small sacrifice in sensitivity and specifity.

Providing pre-determined reference inputs to FCM with desired initial membership values could lead better performance but this study requires more investigation with EEG specialists.

This study shows that fuzzy c-means clustering algorithm with pre-classification stage has promising results. Results of this study may be improved by;

- Increasing number of input samples
- Introducing frequency domain parameters to feature extraction part
- Introducing effect of multiple channels of EEG recordings
- Different methods for pre-processing stage (Fuzzy logic, multi layer artificial neural network structures etc.) to improve high false positive rate
- Different window sizes for spikes when using raw data as input

User of the method developed in this thesis must be careful against the low selectivity performance of the system. It could mislead to incorrect diagnosis of epilepsy as low selectivity means some non-epileptiform spikes in the EEG are incorrectly labeled as epileptiform. Before employing this method in clinical studies, the false detection rate should be decreased.

Another issue that must be considered in EEG records is the effect of body's self electrical activity other than that of the brain. Some studies in epileptiform spike detection field employ algorithms to remove the effects of electromyogram (EMG –

electrical activity measured from the muscles), electroocculogram (EOG - electrical activity measured from the eyes) and electrocardiogram (ECG/EKG - electrical activity measured from the heart) signals from the EEG. For example, muscle activity or eye blinking results in spike-like waves in EEG and therefore increases the number of false positives (Ozdamar, & Kalayci, 1997). To remove artifacts due to ECG, EOG and EMG, many methods are proposed depending on whether a reference ECG, EOG or EMG signal is present or not (Celka, Boashash, & Colditz, 2001)

Fuzzy c-means clustering is a very powerful tool for dealing with noisy and unlabelled data. Each input data is assigned a random membership or each cluster is assigned a random centroid at the initialization of this algorithm. However, one can choose to assign specific membership values or centroids at the initialization of the algorithm to achieve better performance. Two examples of centroid assignment methods for FCM are 'Mountain Clustering Method' and 'Subtractive Clustering Method' (Jang, Sun, & Mizutani, 1997).

The epileptiform spike detection algorithm proposed in this thesis requires all the input data to be ready before the processing starts. That's because of the batch processing nature of fuzzy c-means clustering algorithm. Therefore the method proposed in this thesis can not employed online, i.e, during the real time recording of the EEG. After the recording process is finished, data can be presented to the system for detection of epileptiform activity in the EEG. There exist methods, mostly based on Artificial Neural Networks which may be used online with the recording (Acır, 2004).

A novel approach for diagnosis of epilepsy based on epileptiform spike detection from EEG recordings is proposed in this thesis. When compared with other approved methods, low selectivity is a disadvantage of the system; however it might be improved by further tuning of pre-classification stage. On the other hand, high selectivity implies high detection rate of epileptiform spikes and proposes a new field for researches in epileptiform spike detection.

References

Acır, N. (2004). *Automatic spike detection in EEG*, PhD Thesis, Electrical Electronics Engineering Department, Dokuz Eylül University, Izmir.

Betts, T. (1998). What is epilepsy? *Electrical Engineering and Epilepsy: A Successful Partnership.* Savoy Place, London: IEE

Bezdek, Ehrlich, & Full (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences, 10.*

Bowden, R. (2000). *Learning non-linear models of shape and motion*, PhD Thesis, Department Systems Engineering, Brunel University, Uxbridge, Middlesex, UK

Celka, P., Boashash, B., & Colditz, P. (2001). Engineering in medicine and biology, Preprocessing and time frequency analysis of newborn EEG seizures. *IEEE Engineering in Medicine and Biology, 20.* (5) 30-38

Dayhoff, J. E. (1990). *Neural network architectures: An introduction.* Australia: Van Nostrand Reinhold.

Dingle, A. A., Jones, R. D., Carroll, G. J., & Fright, W.R. (1993). A multistage system to detect epileptiform activity in the EEG. *IEEE Transactions Biomedical Engineering, 40.*

Fraser, N. (1998). *The biological neuron.* Retrieved July 02, 2004 from http://vv.carleton.ca/~neil/neural/neuron-a.html

Holmes, G. L. Classification of seizures and the epilepsies. In Schachter, S.C., & Schomer, D.L., (Eds). (1997). *The Comprehensive Evaluation and Treatment of Epilepsy.* San Diego, CA: Academic Press

Inan, Z. H. (2001). *Simulation of control of a two segment robot arm by artificial neural networks and fuzzy control.* License graduate project, Electrical Electronics Engineering Dept, Dokuz Eylül University, Izmir

James, C. J., Jones, R. D., Bones, P. J., & Carroll, G. J. (1999). Detection of epileptiform discharges in the EEG by a hybrid system comprising mimetic, self-organized artificial neural network, and fuzzy logic stages. *Clinical Neurophysiology, 110,* 2049-2063

Jang, S.R., Sun, T., & Mizutani, E. (1997). *Neuro-Fuzzy and soft computing*. Matlab Curriculum Series, Prentice Hall

Jasper, H. H. (1958). The 10-20 electrode system of the International Federation, report of the committee on methods of clinical examination in electroencephalography. *Electroencephalography Clinical Neurophysiology. 10*,371-375

Karayiannis, N., & Muralidhar, R. (1995). An integrated approach to fuzzy learning vector quantization and fuzzy k-means clustering. *Artificial neural networks in engineering*

Kiloh, L. G., McComas, A. J., Osselton, J. W., & Upton A. (1981) *Clinical electroencephalography* (4th ed). London UK: Butterworths,

Minasny, B., & McBratney, A.B. (2002). *FuzME version 3.0*, Australian Centre for Precision Agriculture, The University of Sydney, Asutralia. http://www.usyd.edu.au/su/agric/acpa.

Niedermeyer, E., & Silva, F.L.D. (1993). *Electroencephalography, basic principles, clinical applications and related fields,* (3rd ed). Baltimore, MD: Williams & Wilkins

Özdamar, O., & Kalayci, T. (1995). Wavelet preprocessing for automated neural network detection of EEG spikes, *IEEE Engineering in Medicine and Biology.*

Özdamar, O., & Kalayci, T. (1997). Detection of spikes with artificial neural networks using raw EEG. *Computer and Biomedical Research, 31*. 122 - 142.

Pang, C. C., Upton, A. R. M., Shine, G., & Kamath, M.V. (2003). A comparison of algorithms for detection of spikes in the electroencephalogram. *IEEE Transactions Biomedical Engineering*, *50*(4), 521-526

Pardey, J., Roberts, S., & Tarassenko, L. (1996). A review of parametric modeling techniques for EEG analysis. *Medical Engineering and Physics. 18* (1). 2-11.

Snell, R. S. (2001). *Clinical Neuroanatomy for Medical Students*, (2nd Ed). Williams & Wilkins

Webber, W.R.S., Litt, B., Wilson, K., & Lesser, R. (1994). Practical detection of epileptiform discharges (EDs) in the EEG using an artificial neural network: a comparison of raw and parameterized data. *Electroencephalography Clinical Neurophysiology, 91.* 194-204

APPENDIX : EEG SPIKE DETECTION WITH FUZZY C-MEANS CLUSTERING SOFTWARE

nn.m

```
user_data='a';    %For entering to while loop we assign to any string
while user_data~='t' & user_data~='e'
  user_data=input('Train(t)  or  Test(e) \n','s');
end

switch user_data
    case 't',
    load data.txt;
    input_data=data';
    load des.txt
    dsr=des;
    case 'e'
    load test_data.txt;
    input_data=test_data';
    load test_des.txt
    des=test_des;
    clear A Q Result Result2 d d2 dsr dn dp mn mp
    clear in_sample pos_indis rev_d2 veys veys2 x1 x2 x3
end

in_size=size(input_data);
for i=1:in_size(1)
   mean=sum(input_data(1,:)/in_size(2));
   A(1,:)=input_data(1,:) - mean;
end

[mp_sp,mn_sp,dp_sp,dn_sp,des_sp]=PrepSampleIn;
[mp,mn,dp,dn,pos_indis]=extract(A);
[d]=form_desired(des,pos_indis,mp);

%--------------------------------------------------------------
switch user_data
    case 't',
```

```
x3(1,:)=[mp mp_sp];
x3(2,:)=[mn mn_sp];
x3(3,:)=[dp dp_sp];
x3(4,:)=[dn dn_sp];
ones1=-10*ones(1,size(x3,2));  % Network bias
x3(5,:)=[ones1];

d2(1,:)=[d des_sp];
n=0.0005;        %Learning Coefficent
veys=rand(5,1);     %Initial Weight Vector
veys2=rand(5,1);     %Initial Weight Vector
rev_d2=bitcmp(d2,1);  %Desired output for second neuron
%------------------------------------------------------------------
if 1
  rnd=[randperm(size(x3,2))];
  size_rnd=size(rnd);
  while size_rnd(2)<30000      %Learning loop must be least 10000
    rnd=[rnd randperm(size(x3,2))];
    size_rnd=size(rnd);
  end
  for j=1:size_rnd(2), %It takes inputs randomly
   in_sample=rnd(j);
   x1=[x3(1,in_sample);x3(2,in_sample);x3(3,in_sample);x3(4,in_sample);ones1(in_sample)];
   %Calculate first perceptron output
   Q(j)=hardlim(veys'*x1);
   veys=veys+n.*(d2(in_sample)-Q(j)).*x1; %Calculate new weights
   %Second neuron
   Z(j)=hardlim(veys2'*x1);
   veys2=veys2+n.*(rev_d2(in_sample)-Z(j)).*x1;
  end
  %save weights
  w=veys;  w2=veys2;
  % Test trained network
  for j=1:(size(x3,2))
    x2=[x3(1,j);x3(2,j);x3(3,j);x3(4,j);ones1(j)];
    Result(j)=hardlim(w'*x2);  %Products of weight and inputs
  end
  for j=1:(size(x3,2))
    x2=[x3(1,j);x3(2,j);x3(3,j);x3(4,j);ones1(j)];
```

```
        Result2(j)=hardlim(w2'*x2);   %Products of weight and inputs
      end
else % Matlab nn toolbox implementation
      [net,trrec,out,Err,Y]=prcptrn(x3,d2);
      [net2,trrec2,out2,Err2,Y2]=prcptrn(x3,bitcmp(d2,1));
      % Test trained network
     Y=sim(net,x3);
     Y2=sim(net2,x3);
end

case 'e',
    x3(1,:)=[mp];
    x3(2,:)=[mn];
    x3(3,:)=[dp];
    x3(4,:)=[dn];
    ones1=-10*ones(1,size(x3,2));   % Network bias
    x3(5,:)=[ones1];

    d4=d(:,1:size(mp,2));
    if 1
       for j=1:size(x3,2)
          Result(j)=hardlim(w'*x3(:,j));   %Products of weight and inputs
       end
       %weights of second neuron:w2  desired vector:rev_d2 results:Result2
       rev_des=bitcmp(d4,1);
       for j=1:size(x3,2)
          Result2(j)=hardlim(w2'*x3(:,j));   %Products of weight and inputs
       end

%Save features of 3rd group of spikes!
third=find(Result==1 & Result2==1);
third_features=x3(1:4,third);

save third.txt third_features -ascii

    else  % Matlab nn toolbox implementation
       Y=sim(net,x3);
       Y2=sim(net2,x3);
    end
```

```
end
```

prcptrn.m

```
function[net,trrec,out,Err,Y]= prcptrn(Xin,d)

mat=[0 150; 0 150; 0 50; 0 50; -10 0];

%NPN is more time efficient for inputs with high variance.
net=newp(mat,1,'HARDLIM','LEARNPN');
in=Xin;
des=d;

Y=sim(net,in);
net.trainParam.epochs = 1000;
net.trainParam.goal = 0.0005;
[net,trrec,out,Err] = train(net,in,des);
Y=sim(net,in);
```

Extract.m

```
function[mp,mn,dp,dn,pindis]= extract(A)

data_length=size(A,2);
for J = 1:data_length-1     %Taking the differences between indices
  B(1,J) = A(1,J+1)-A(1,J);
end

delta=hardlim(B);
y=size(B);  %Size of difference vector
%%%% Find peaks
for J = 1:y(2)-1,     %If there are some indices in one direction
  if delta(1,J)==delta(1,J+1)     %it eliminate elements between first and last element
    peak(1,J+1)=0;
  else
    peak(1,J+1)=1;
  end
end

peak(1)=1;
```

```
count=0;
for K = 1:y(2)-1    %New triangular input vector is indis
  if peak(1,K)==1  % There is a peak
    count=count+1;  % No of peaks
    indis(1,count)=K;  % Indices of peaks
  end
end

for I = 1:size(indis,2)-1,    %Difference between indices
  if indis(1,I+1)~=0
     if A(1,indis(1,I+1))==0
        peak_diff(1,I)=0;
     else
        peak_diff(1,I) = A(1,indis(1,I+1))-A(1,indis(1,I));
     end
  end
end

%If the peak is smaller than preceeding and following element
%and less than threshold it is eliminated. New input vector is indis2
count=0;
for I = 2:size(indis,2)-2,
  if    abs(peak_diff(1,I+1))<abs(peak_diff(1,I))    |    abs(peak_diff(1,I))>abs(peak_diff(1,I-1))    |
abs(peak_diff(1,I))>0.5;
     count=count+1;
     indis2(1,count)=indis(1,I+1);
  end
end

indis2(1)=indis(1);
indis2(2)=indis(2);
p=size(indis2);    %Size of new input vector

for I = 1:p(2)-1,    %Taking the differences between indices
  if indis2(1,I+1)~=0
    if A(1,indis2(1,I+1))==0
      C(1,I)=0;
    else
      C(1,I) = A(1,indis2(1,I+1))-A(1,indis2(1,I));
```

```
    end
  end
end

A3=hardlim(C);
z=size(C);    %Size of difference vector

for J = 1:z(2)-1,    %If there are some indices in one direction
  if A3(1,J)==A3(1,J+1)    %it eliminate elements between first and last element
    A4(1,J+1)=0;
  else
    A4(1,J+1)=1;
  end
end

A4(1)=1;
count=zeros(9,1);
for K = 1:p(2)-2,    %New triangular input vector is indis1
   if A4(1,K)==1
      count=count+1;
      indis1(1,count)=indis2(1,K);
   end
end

Y=size(indis1);
% collect positive indices
my_ind=1;
for y=2:1:Y(2)
   if indis1(1,y)~=0
      if (A(1,indis1(1,y))-A(1,indis1(1,y)-1))>0
         pindis(1,my_ind)=indis1(1,y);
         my_ind=my_ind+1;
      end
   end
end

size_sp=size(pindis);

for k=1:size_sp
```

```
  raw_data(k,:)=A(pindis(1)-10:pindis(1)+30);
end
save raw_data.txt raw_data -ascii

% Her bir positive spike için bir önceki neg spike ile aralık (dp), bir sonraki
% neg spike ile aralık (dn), ve amp farkları bulunacak.
for y=1:size_sp(2)-1
  if pindis(1,y)~=0
    [k l]=find(indis1(1,:)==pindis(1,y));
      dp(1,y)=indis1(1,l)-indis1(1,l-1);
      dn(1,y)=indis1(1,l+1)-indis1(1,l);
      mp(1,y)=A(1,indis1(1,l))-A(1,indis1(1,l-1));
      mn(1,y)=abs(A(1,indis1(1,l+1))-A(1,indis1(1,l)));
  end
 end
```

form_desired.m

```
function[d]= form_desired(dsr,pindis,mp)
des=[];
% Process desired
for y=1:size(dsr,2)
  if dsr(1,y)~=0
    [g,h]=find(pindis(1,:)==dsr(1,y));
    if(isempty(h))
      des(1,y)=0;
    else
      des(1,y)=h;
    end
  end
end
d(1,:)=zeros(1,size(mp,2));          %Desired is 1 ohterwise is 0

for j=1:size(des,2)
  if des(1,j)~=0
    d(1,des(1,j))=1;
  end
end
```

get_test_data.m

```
des =[];
data_ch1=[];

%-----------------------------------------------
%1033.txt   4 SPIKES
%-----------------------------------------------
load data_1033.txt
data=data_1033*-1;

data_ch1(:,1) = data(1:4000,9); %CHANNEL 9
des(1)= 2124;
des(2)=2181;
des(3)= 2603;
des(4)= 3584;

raw1=data_ch1((2124-10):(2124+30),1);
%-----------------------------------------------
%1005.txt   5 SPIKES
%-----------------------------------------------
load data_1005.txt
data=data_1005*-1;

data_ch1 = [data_ch1; data(1:3000,9)]; %CHANNEL 9
des(5)= 4676;
des(6)=5099;
des(7)= 5546;
des(8)= 6757;
data_ch1 = [data_ch1; data(1:1000,15)]; %CHANNEL 9
des(9)= 7675;

%-----------------------------------------------
%1008.txt   6 SPIKES
%-----------------------------------------------
load data_1008.txt
data=data_1008*-1;

data_ch1 = [data_ch1; data(1:9000,9)]; %CHANNEL 9
des(10)= 10673;
```

```
des(11)=13325;
des(12)= 13701;
des(13)= 15805;
des(14)= 16019;
des(15)= 16272;

save test_data.txt data_ch1 -ascii
save test_des.txt des –ascii
```

PrepSampleIn.m

```
function [mp_sp,mn_sp,dp_sp,dn_sp,des_sp] = PrepSampleIn()

%1042.txt 9 spikes
mp1=[43.1 55.5 78.8 89.2 84.7 65.3 57.6 97];
mn1=[85.7 77.1 132.3 119.3 104 114.1 110.7 162.9];
dp1=[18 11 8 9 18 13 19 12];
dn1=[16 15 15 17 18 18 18 16];

%987.txt
mp2=[77 41.5 80.7 107.5 68.1 93.8 49.8 59.9];
mn2=[53.5 73.7 114.7  119.7  79.4  113.9 63  84.5];
dp2=[5 4 5 9 18 10 13 4];
dn2=[6 9 5 12 14 12 12 6];

%1038.txt
mp3=[265.7 141.2 41.9 194 144.2 19.6 236.2 113.5 108.7 153  191.8 4.4 138.5 139.3 125.7 45.5
108.6];
mn3=[119.6 65 74.5 81 18.1 67.2 63.4 150.1 159.6 130.7 101.1 32.7 56.9 106.7 30.3 56.1 101.3];
dp3=[30 21 6 26 19 4 21 20 11 17 28 3 25 19 25 10 16];
dn3=[23 13 7 13 4 6 14 23 23 25 19 5 5 24 6 20 18];

%1007.txt
mp4=[59.2 92.7 71.4 72.3 82.9 41.3 114.4];
mn4=[85.5 100.5 104.1 97.4 89.4 94.7 181.3];
dp4=[12 12 17 14 12 10 1];
dn4=[16 21 20 22 19 23 20];

mp_sp=[mp1 mp2 mp3 mp4];
mn_sp=[mn1 mn2 mn3 mn4];
```

```
dp_sp=[dp1 dp2 dp3 dp4];
dn_sp=[dn1 dn2 dn3 dn4];
des_sp=ones(1,40);
```

fuzzy.m

```
%Fuzzy c-means clustering for epilepti-form
%spike detection from eeg signals
clear all;

load in_form.txt             % Load data from a text file
%load raw_all.txt              % Load data from a text file
data = in_form';             % data to be clustered
%data = raw_all;               % data to be clustered
load third.txt            % Load data from a text file
data2= third';              % data to be clustered

data=[data2;data];

%Set parameters
ndata = size(data, 1);      % number of data
ndim = size(data, 2);       % number of dimension

%Set parameters-2
phi=2;                  % fuzzy exponent >1
maxiter=50000;              % maximum iterations
toldif=0.000001;          % convergence criterion
scatter=0.2;             % scatter around initial membership
ntry=10000;               % number of trial to choose an optimal solution

N_clusters=2; %Number of clusters

[center, U, obj_fcn] = fcm(data, N_clusters,[phi,maxiter,toldif,1]);

maxU = max(U);
index1 = find(U(1, :) == maxU);
index2 = find(U(2, :) == maxU);

for i=1:size(index1,2)
   if U(1,index1(i))>0.55
```

```
    m1(i)=index1(i);
  end
end

for i=1:size(index2,2)
  if U(2,index2(i))>0.55
    m2(i)=index2(i);
  end
end

%compare
load des_form.txt
des = des_form;
des_ind=find(des==1);
```