Liver Illness Diagnosis Based on CMAC (Cerebellar Model Articulation Controller) Neural Network Approach

by

Semra Baki

A thesis submitted to

the Graduate Institute of Sciences and Engineering

of

Fatih University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Engineering

February 2009
Istanbul, Turkey

APPROVAL PAGE

Student : Semra Baki

Institute : Institute of Sciences and Engineering

Department : Computer Engineering

Thesis Subject: Liver Illness Diagnosis Based on CMAC

(Cerebellar Model Articulation Controller) Neural Network Approach

Thesis Date : February 2009

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Assist. Prof. Tuğrul YANIK
Head of Department

This is to certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist. Prof. İhsan Ömür BUCAK
Supervisor

Examining Committee Members

Assist. Prof. İhsan Ömür Bucak …….

Prof. Dr. Bekir Karlık …….

Prof. Dr. Sadık Kara …….

It is approved that this thesis has been written in compliance with the formatting rules laid down by the Graduate Institute of Sciences and Engineering.

Assist. Prof. Nurullah ARSLAN
Director

Liver Illness Diagnosis Based on CMAC
(Cerebellar Model Articulation Controller) Neural Network Approach

Semra Baki

M. S. Thesis – Computer Engineering
February 2009

Supervisor: Assist. Prof. İhsan Ömür Bucak

ABSTRACT

The liver is one of the most important organs of the human body because of its vital functions. If the liver malfunctions in anyway, people know that they are putting their life at risk. For this reason, diagnosing any disease in the liver is important and sometimes difficult. It is also important to notice the diagnosis of the patient at an early stage as the symptoms arise so that the patient might be able to carry on a normal life. In the diagnosis of the disease, the physician can run a liver function test, a urine test, and other comparable tests to test the liver enzyme and assess the phase of the disease. The objective of this thesis is to diagnose the liver disease using an application of the CMAC (Cerebellar Model Articulation Controller) neural network.

Keywords: Liver disease, Liver enzymes, CMAC neural networks, Medical diagnosis.

CMAC Yapay Sinir Ağı ile Karaciğer Hastalıklarının Teşhisi

Semra Baki

Yüksek Lisans Tezi – Bilgisayar Mühendisliği
Şubat 2009

Tez Yöneticisi: Yrd. Doç. Dr. İhsan Ömür Bucak

ÖZ

Karaciğer, gerçekleştirdiği fonksiyonlar dolayısı ile insanın hayati organlarından biridir. Karaciğerin bu fonksiyonlardan herhangi birini yerine getirememesi insan hayatını tehlikeye atma anlamına gelir. Bu sebeple karaciğer hastalıkları teşhisi oldukça önemli ve bazen bir o kadar da zahmetli olmaktadır. Teşhis aşamasında hastada var olan semptomların erken farkına varılması, hastalığın ilerleyip daha ciddi bir boyuta ulaşmaması için önem arz eder. Hastalığın teşhisinde, karaciğer fonksiyon testi, idrar testi gibi testler uygulanarak bu testlerde ki karaciğer enzimleri değerlendirilip hastalığa ve evresine karar verilmektedir. Bu tezde CMAC (Cerebellar Model Articulation Controller) tabanlı sinir ağı kullanılarak karaciğer hastalıklarının teşhisini sağlamaya yönelik bir uygulama gerçekleştirilmiştir.

Anahtar Kelimeler: Karaciğer Hastalıkları, Karaciğer Enzimleri, CMAC Yapay Sinir, Tıbbi Tanıma.

To my parents,

ACKNOWLEDGEMENT

I would like to thank all the people who directly or indirectly collaborated in the completion of this thesis work. First of all, I would especially like to thank my thesis supervisor Assist. Prof. İhsan Ömür Bucak for his guidance throughout the research.

I would like to thank my jury members, Prof. Dr. Bekir Karlık and Prof. Dr. Sadık Kara for their valuable suggestions and comments. I would also like to thank MD. Yavuz Pehlivan for providing us with the clinical data.

I express my thanks and appreciation to my family for their advice and patience. Moreover, I want to thank my intended, Musa Baki, for always helping me and being such a good influence. I also thank my friends for their motivation.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

LIST OF SYSMBOLS AND ABBREVIATIONS

SYMBOL/ABBREVIATION

ALAP	Alkaline Phosphatase
ALT	Alanin Aminotransferaz
ANN	Artificial Neural Network
AST	Aspartat Aminotransferaz
CMAC	Cerebellar Model Articulation Controller
CT	Computerized Tomography
GGT	Gamma-Glutamyl Transpeptidase
HAV	Hepatitis A
HBV	Hepatitis B
HCV	Hepatitis C
LAP	Leucine Aminopeptidase
LMS	Least Mean Square
PLT	Pletalet
PROT	Protein

CHAPTER 1

INTRODUCTION

Medical diagnosis process realized by using Articial Neural Networks illustrates a very good way a combination of technology and medicine. In this study, we have tried to bring an approach to the liver disease by the technological perspective.

This study aims to develop an illness diagnosis system for liver cells. Neural network is exactly suitable for diagnosis process with many different applications and worldwide.

1.1 Liver and Liver Illness Based on CMAC Neural Network

The liver is one of the most important organs in the human body that lies below the diaphragm (Hopkins, 2008). The liver has both largest gland and largest internal organ in the human body (Petska, 2007). It performs several numbers of metabolic functions that are essential to human life.

Some of the more well-known functions of the liver (Darwin, 2008), and (McLaughlin, 2000):

- Production of bile, which helps carry away waste and break down fats in the small intestine during digestion,
- production of certain proteins for blood plasma,
- production of cholesterol and special proteins to help carry fats through the body,

- conversion of excess glucose into glycogen for storage. This glycogen can later be converted back to glucose for energy,

- regulation of blood levels of amino acids, which form the building blocks of proteins,
- processing of hemoglobin for use of its iron content (The liver stores iron.),
- conversion of poisonous ammonia to urea (Urea is one of the end products of protein metabolism that is excreted in the urine.),
- clearing the blood of drugs and other poisonous substances,
- regulating blood clotting,
- resisting infections by producing immune factors and removing bacteria from the blood stream.

If the liver cannot do the things we mentioned above, there might arise various diseases. There are diseases that occur in liver in short time (acute) and long time (chronic) period. These diseases could occur because of medications, alcohol, viruses or fat. Some of these disease are the inflammation of the liver, insufficient liver performance, Hepatitis A, B, C, D and liver cirrhosis.

Hepatitis is the inflammation of the liver, resulting in liver cell damage and destruction (Adams, B., 2008). Hepatitis diseases are caused by hepatitis viruses. Hepatitis viruses have six main types (Adams, B., 2008). The viruses identified until now have been named as A, B, C, D, E, and G, from which A and E are contagious (Dimitriou, 2008).

Hepatitis A virus (HAV) is heat stable and will survive for up to a month at ambient temperatures in the environment (Gott, 2008).

Hepatitis B virus (HBV) can cause lifelong infection, cirrhosis (scarring) of the liver, liver cancer, liver failure, and death. Hepatitis B vaccine is available for all age groups to prevent Hepatitis B virus infection (Gott, 2008).

Hepatitis C virus (HCV) is spread by contact with the blood of an infected person. The progression of Hepatitis C is typically silent until it is late in the course of disease. When symptoms are present, often liver cirrhosis (scarring) has been occurred. The progression to liver cirrhosis only occurs in 20 percent of those with Hepatitis C and

liver failure develops in about 5 percent. From the time of acquisition of the virus it can take up to 50 years to develop cirrhosis. Most people with Hepatitis C are unaware that they have it (Gott, 2008).

Hepatitis D can only occur in the presence of Hepatitis B. If an individual has Hepatitis B and does not show symptoms, or shows very mild symptoms, infection with Hepatitis D can put that person at risk for full-blown liver failure that progresses rapidly. Hepatitis D can occur at the same time as the initial infection with B, or it may show up much later. Transmission of Hepatitis D occurs the same way as Hepatitis B, except that the transmission from mother to baby is less common (Adams, B., 2008).

Hepatitis E is similar to Hepatitis A. Transmission occurs through fecal-oral contamination. It is less common than Hepatitis A. Hepatitis E is the most common in poorly developed nations and rarely seen in the United States. There is no vaccine for Hepatitis E at this time (Adams, B., 2008).

Hepatitis G is the newest strain of hepatitis and very little is known about it. Transmission is believed to occur through blood and is most commonly seen in drug users, individuals with clotting disorders such as hemophilia, and individuals who require hemodialysis for renal failure (Adams, B., 2008).

Fatty liver is excessive accumulation of fat inside the liver cells. Fatty liver is the most common alcohol-induced liver disorder. The liver is enlarged, causing upper abdominal discomfort on the right side (Darwin, 2008).

Liver cirrhosis is a term that refers to a group of chronic diseases of the liver in which normal liver cells are damaged and replaced by scar tissue. Liver cirrhosis does not develop overnight. It takes several years to develop. There are usually no symptoms while liver cirrhosis is developing. Symptoms usually appear when liver cirrhosis is fully developed. The symptoms will depend on how severe liver cirrhosis is (Gott, 2008). The list of diagnostic test cirrhosis of the liver includes physical exam, liver function tests, computerized axial tomography scan, and liver biopsy (Icer and Kara, 2006).

Liver tumors are abnormal masses of tissue that form when cells begin to reproduce at an increased rate. The liver can grow both noncancerous (benign) and cancerous (malignant) tumors. Often they are not diagnosed until an ultrasound, CT (computerized tomography) scan, or MRI is performed (Darwin, 2008).

When diagnosing liver disease, the physician looks at the patient's symptoms and conducts a physical examination. In addition, the physician may request a liver biopsy, liver function tests (AST, ALT, biluribin), an ultrasound, or a CT scan (Darwin, 2008). Liver function tests can diagnose viral hepatitis and autoimmune liver diseases. An ultrasound scan will show blockage of the bile duct, fatty liver, cirrhosis and liver tumors (Babe, 2007).

Diagnosing a liver problem can be a difficult task, because symptoms do not often appear until the later stages of most liver diseases and conditions. By then the liver may have suffered serious or permanent damage.

Liver disease diagnosing process can be difficult by that time the liver may have suffered critical or permanent damage, because symptoms of liver disease do not often appear the later stages of most liver diseases and conditions (Dupage, 2007).

When symptoms do begin to appear, they might include (Dupage, 2007):

- Irregular sleep, including a tendency to sleep at odd hours and wake up in the middle of the night,
- low or fluctuating energy levels. Lows tend to come in mid morning and mid afternoon,
- losing weight becomes even more difficult than usual, even though you eat smarter and start exercising,
- your skin and eyes will start to have a yellowish tint. Other skin problematic conditions like acne, eczema, psoriasis, and general itchiness may appear,
- You may experience bad reactions to drugs and medications such as headache pills, antibiotics and anti-histamines,

- drinking even small amounts of alcohol can make you feel inebriated. You will feel intense hangovers when you drink,
- caffeine will have a much stronger effect and could keep you awake for hours,
- you may experience digestion problems, especially with creamy, oily and fatty foods. You may even feel nauseous after you eat them,
- you have stomach bloating and gas more often than usual,
- eating asparagus will cause a strange smell in your urine. This is a classic liver disease symptom,
- you will occasionally experience a warm, flush feeling will start in your trunk and rise upward toward the head,
- you will begin to get frequent headaches, heartburn and acid reflux,
- certain substances may cause a severe reaction, especially cleaning products, gasoline, paint, perfumes, bleaches, and so on.

In this thesis, we provide an alternative way to medical diagnosis. The physician may spend very long time for the assessment of the enzyme numbers during normal diagnostic period while making a decision based on those enzymes. This study provides a contribution to the medical diagnosis process by shortening the time through the use of an intelligent model and helps the physician to diagnose complex cases which are otherwise difficult to perceive. Physicians make a decision according to enzyme values in normal diagnosis stage of this method .

A multilayer neural network is used as one of the most popular methods for diagnostic processes. In this paper, the CMAC neural network has been preferred over the multilayer neural network. Because the multilayer neural network requires many iterations and a large number of computations per iteration to converge an output so that the algorithm runs slowly (Miller et all, 1990). However, the CMAC presents many attractive features that are useful for real time applications. The CMAC has been used to solve various robotic problems, control applications, medical applications, pattern recognition, signal processing and image processing applications (Albus, 1975a), (Albus, 1975b), and(Miller, Glanz, and Kraft, 1990). For example Hormel has applied Kohonen-type algorithms to adapt the storage mechanisms of the CMAC to match the input distribution (Hormel, 1990). Campanga (Kraft, and Campanga, 1990) have

compared a CMAC-based controller with two traditional adaptive controllers. Bucak (Bucak, 2008), and (Karlık, and Bucak, 2009) has applied a diagnosis process such as odor recognition and detection of drinking water quality with CMAC algorithm. Chin-Pao and others (Hung et all, 2003) have investigated the fault tolerance of CMAC Networks. Chin-Pao Hung has applied a diagnosis process such as PIC microcontroller based fault diagnosis apparatus design for water circulation system using CMAC neural network approach (Hung et all, 2007).

In addition to CMAC research, there is a remarkable research on prediction of liver disease. In this research, Zhu and others have investigated prediction of radiation induced liver disease using artificial neural network (Zhu et all, 2006). This research shows an example of identification process of liver diseases by using neural Networks.

In Chapter 1, we describe liver, liver disease and its symptoms. It also discusses why we chose the CMAC Neural Network algoritm for diagnosing of the liver disease. Chapter 2 gives a detailed discussion of the CMAC neural network and its significant properties. Chapter 3 discusses the CMAC based liver diagnostic system which covers all the steps used in this thesis such as pattern collection, the CMAC liver diagnostic system, training data, quantization process, learning rule, learning convergence and performance evaluation, diagnosis algorithm and case study, and finally discussion. The last chapter summarizes the results and presents future research.

CHAPTER 2

CMAC NEURAL NETWORK

2.1 General Description of CMAC Neural Network

Cerebellar Model Articulation Controllers (CMACs) is firstly proposed during 1970s by James Albus at National Bureau of Standarts. J. Albus's idea was based on a model of cerebellum, which is part of a brain and is responsible for learning process. Albus used the CMAC to do rote learning of movements of an artificial arm (Miller et al, 1990). The CMAC have been popularized by group of professors at the robotics laboratory of the Department of Electrical and Computer Engineering at The University of New Hampshire (Miller et al, 1990).

A general description of the CMAC is that it is a conversion device that converts given input vectors into associated output vectors (Burgin, 1992). The CMAC is an algorithm which quantizes and generalizes its input, produce active memory addresses, and produce an output with summing all the weights in the active memory adresses (Handelman, 1990). This process of finding the output have several steps. Figure 1 shows a CMAC Neural Network with two inputs and one output. This figure describes how to find an output process. According to this figure, there exist four steps to produce the outputs in the CMAC algorithm. Firstly, input state space have one or more input vectors. These vectors re composed of discrete points. These discrete points are connected to the second step of CMAC known as state space detectors. The state space detectors is often called the CMAC's virtual memory (Burgin, 1992). During this transformation input quantization and input generalization processes are carried out.

These process will be explained in detail in the next section. The next step is mapping from the state space detectors into the physical memory. This mapping process

may be realized in two different ways. First one is one-to-one mapping and the other one is many-to-one mapping or random mapping. Once the physical memory is assumed to be smaller than the number of state space detectors, this mapping process is called many-to-one mapping (Burgin, 1992). If the state space detectors are not small enough for one-to-one mapping with physical memory, then random mapping should be used. In other words, if the state space detectors are small enough for one-to-one mapping with pysical memory, we should use one-to-one mapping. The last step includes summing all the weigths in the physical memory to produce the output vectors. During the training, if the output vectors of the CMAC do not match a desired output for a given input state, the weights pointed to by the physical addresses are updated using the following simple steepest-descent update (the least mean square-LMS) rule (Handelman, 1990):

$$w_{j(new)} \leftarrow w_{j(old)} + \beta \frac{(y_d - y)}{g} \tag{2.1}$$

In this update equation, w_j is the weight, y_d, the desired output of the CMAC system, y, the actual output of system, and β, the learning factor. β can take any value between 0 and 1. Values of β, too close to 1, can produce unstable learning behavior in certain situations (Rumelhart and McClelland, 1986). The goal is to find the weights that minimize the error, which is defined as the difference between desired and actual output as indicated earlier as based on the LMS delta rule.

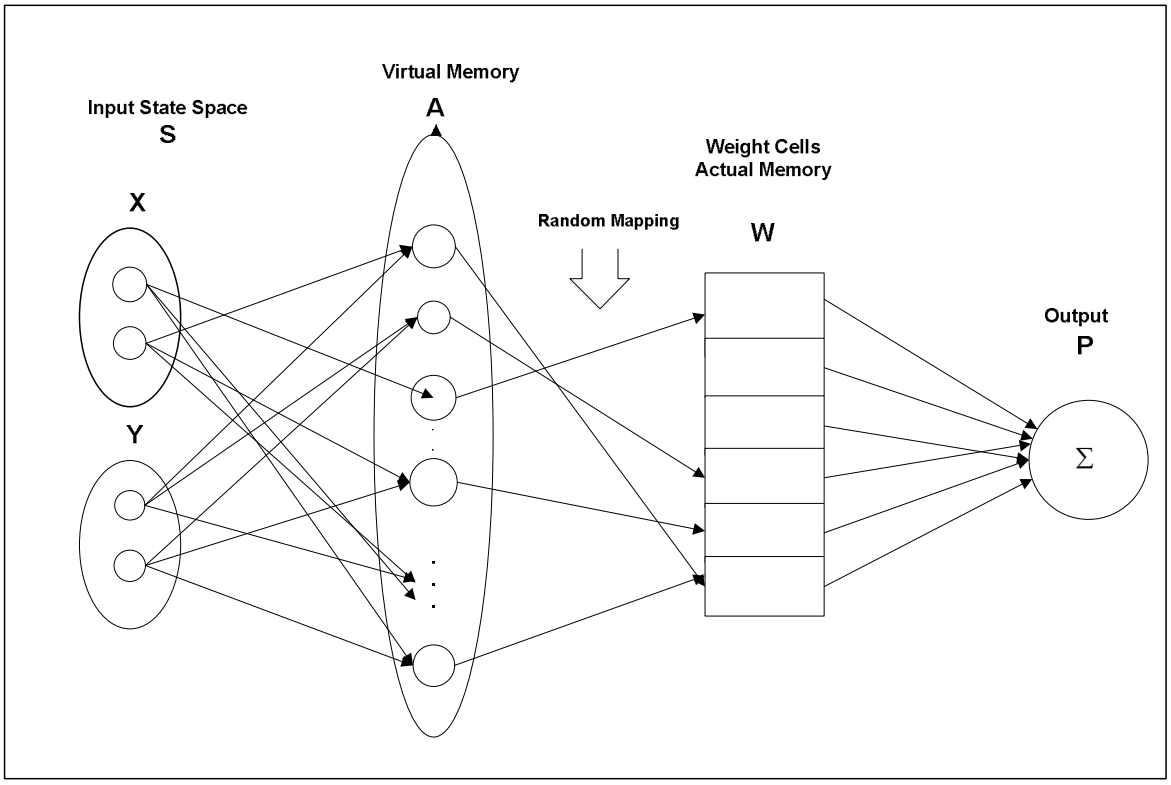

Figure 2.1 A block diagram of CMAC ANN.

Mapping between the input state space and state space detectors (virtual memory) employed by CMAC has such a property that any two input vectors that are similar (close together in input space) will select an overlapping subset of locations in the virtual memory. Thus, the output response of CMAC to similar input vectors will tend to be similar because of the memory locations which are in common. Hence CMAC tends to generalize (Albus, 1975b), and (Burgin, 1992). In this mapping, distant inputs will have a finite probability of sharing some of the same memory locations in virtual memory causing an undesirable generalization. The propability of such collisions depends on the size of the input spate, the generalization parameter *g*, and the size of the virtual memory (Comoglio, 1991). Generalization parameter is the ratio of the width of receptive field (virtual memory) between adjacent layers of receptive fields. The width of the receptive fields produces input generalization, while the offset of the adjacent layers of receptive fields produces input quantization, (Miller, and Glanz, 1996). Generalization parameter determines how many addresses in the virtual memory are excited by the vectors. According to Figure 2.1, *g* is 3, so that all the input vectors overlay 3 distinct locations from the virtual memory.

2.2 Input Quantization in CMAC Neural Network

In this section, we will describe the process of input quantization. In order to store information in memory, dividing the continuous input space into discrete sample intervals is a necessary process in CMAC learning procedure. This process is called quantization (Lu et all, 2006). Quantization concept has been developed due to the fact that the minimum variations in the input values do not affect the output values. During the quantization process, quantization levels affect the values of the input vector, so that the quantization has three levels; each input vector can only assume the three values, such as zero, one or two (Burgin, 1992). The stability of inputs depends on the level of quantization. If the quantization level increases, the stability of inputs increases.

$$q_i = Q(X_i, X_{i\,\min}, X_{i\,\max}, q_{i\,\max}) \qquad i=1,\ldots,m \tag{2.2}$$

The values of each input vector are quantized with equation (2.2), where m is the number of inputs. The resolution of this quantization depends on the expected minimum and maximum input values, $X_{i_{\min}}$ and $X_{i_{\max}}$, and on the number of quantization levels, $q_{i_{\max}}$ (Handelman, 1990).

Figure 2.2 shows a functional schematic of a three-input, single output CMAC module that was designed by Handelman, where the inputs are represented by x_1, x_2, x_3 while the output is by y (Handelman, 1990). The generalization parameter, g is 4. To compute the output, the CMAC algorithm quantizes its inputs, generate active memory addresses, and sum all the weights in the active memory addresses so as to compute the output. Figure 2.2 shows that the mapping Q produces q_1, q_2, and q_3, the quantized versions of the three inputs x_1, x_2, and x_3. The next mapping, V [Eq (2.3)], computes the segments of addresses that, when concatenated [Eq (2.4)], from virtual weight addresses shown v_1, v_2, v_3 and v_4 is composed of three segments, one from each input. The quantization and segment mappings enable the CMAC to generalize, i.e., to produce similar outputs in response to similar inputs. Continuous variants in input values translate into discrete variations in input quantization levels. In the case that an input quantization level changes by 1, the same change occurs only in one of its virtual address segments.

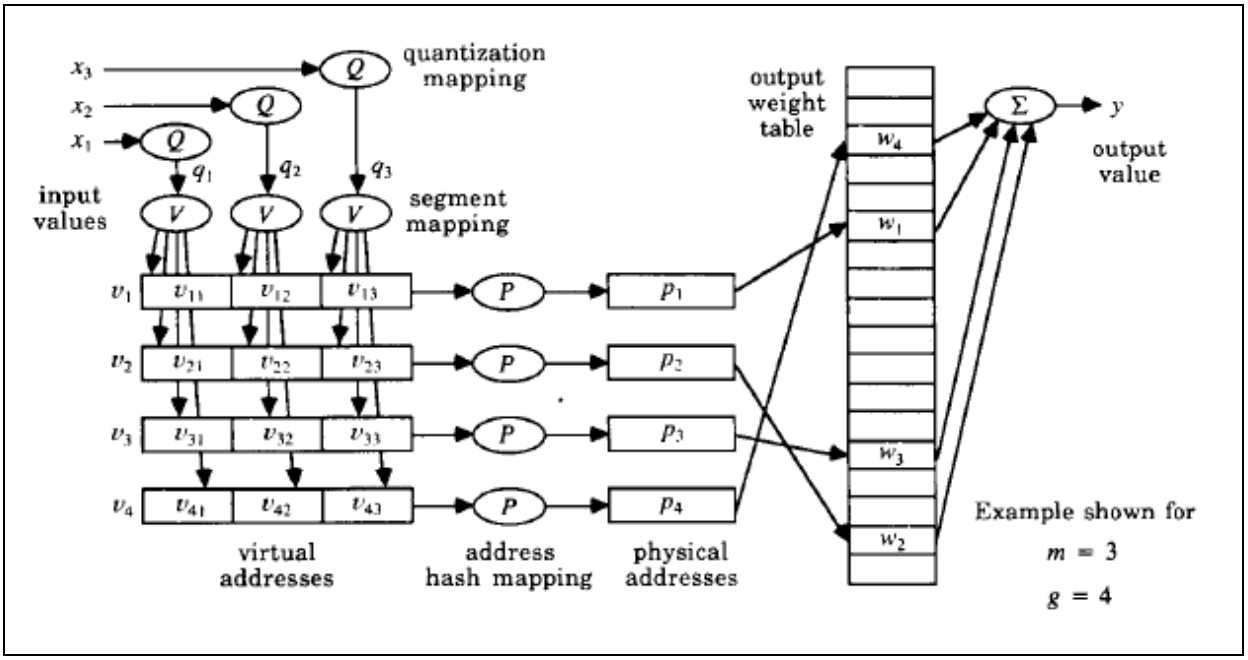

Figure 2.2 Functional schematic of Cerebellar Model Articulation Controller (CMAC) (Handelman, 1990).

$$V_{ji} = V(q_i, g, j), \qquad j = 1,...,g \tag{2.3}$$

$$V_j = concat(V_{j1}, V_{j2}, ...V_{jm}) \tag{2.4}$$

We will, now, summarize the quantization concept with an example to understand better. Assume that the value of input x_2 produces quantization $q_2 = 4$. In this case, Figure 2.3 shows that the virtual address segments associated with this input would be $v_{12} = 5, v_{22} = 6, v_{32} = 7,$ and $v_{42} = 4$. If q_2 shifts from 4 to 5, all virtual address segments remain the same except v_{42} as it shifts from 4 to 8. Consequently, for the network of Figure 2.2 and Figure 2.3, as only one address will have changed (with the assumption that the other two input levels remain constant), outputs associated with neighboring input quantization levels will have three of four virtual weight addresses in common (Handelman, 1990).

The number of weights summed to obtain an output depends on the amount of network generalization, *g*. For a specific input quantization mapping, an increase in *g* means an increase in the amount of shared weights between neighboring input/output

pairs. An increase in the number of quantization levels, $q_{i_{\max}}$, results in higher input resolution, but concurrently increases the size of virtual address space (Handelman, 1990).

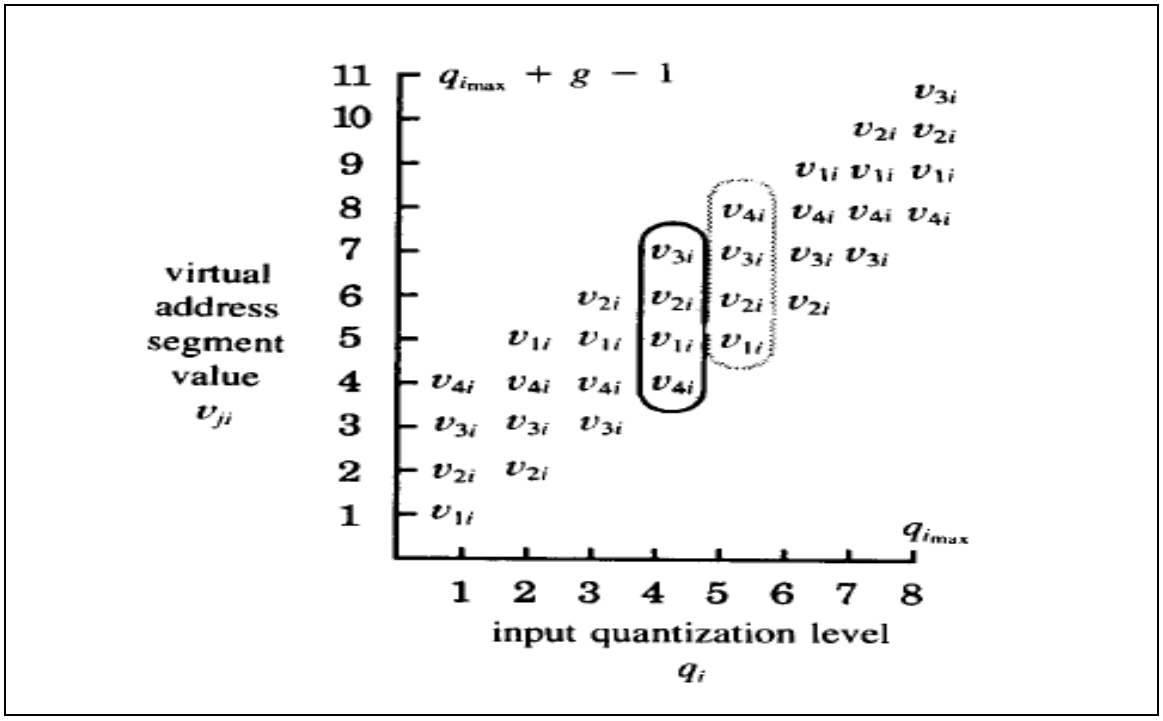

Figure 2.3 CMAC Segment Mapping Function (Handelman, 1990).

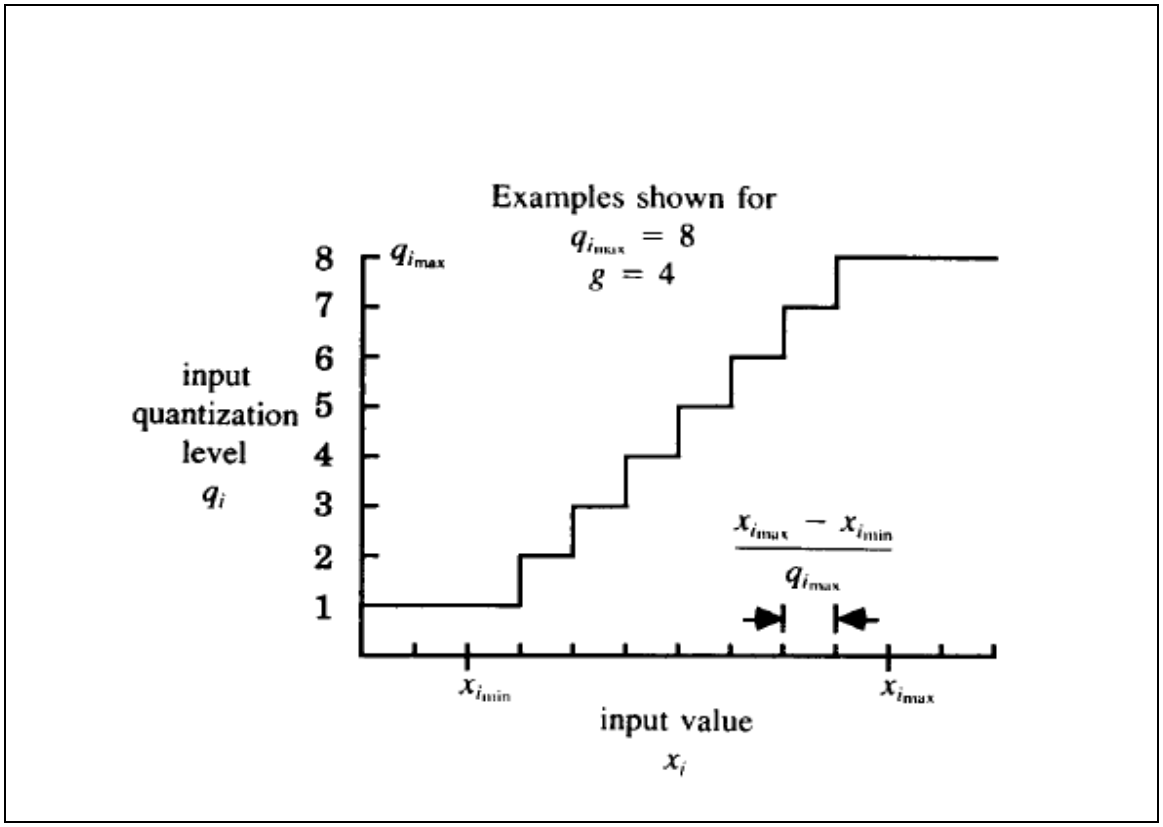

Figure 2.4 CMAC Input Quantization (Handelman, 1990).

2.3 Properties of the CMAC

In this section, we will summarize the properties of the CMAC and mention pros and cons of these properties over the CMAC. The CMAC has several potential advantages over other neural network structures.

The CMAC accepts real inputs and produce real outputs.The input components are quantized, but the number of levels can be as large as desired so that any degree of accuracy is achievable (Miller et all, 1990).

The CMAC has built-in local generalization which means that during mapping between input state and state space detectors, the CMAC has the property that any two input vectors that are similar or close in the input space will select a highly overlapping subset of locations in the state space detectors. Thus, the output response of the CMAC to similar input vectors will tend to be similar because of many memory locations

which are in common. Hence, CMAC tends to local generalization. The amount of generalization depends on the number of overlapping memory locations in the state space detectors (Burgin, 1992).

The CMAC can learn a wide variety of functions. It is easy to show, for example, that a one-input CMAC can learn early any discrete one-dimensional single-valued function, given a few mild conditions on the parameters of the CMAC (Miller et all, 1990).

The CMAC calculations are summations of output memory locations instead of multiple calculations per iteration.The time required per iteration will be much smaller with the CMAC than that of backpropagation. Therefore, given the same number of iterations, the CMAC will learn faster than backpropagation (Comoglia, 1991). The CMAC may actually take fewer iterations than multilayer perceptrons for certain problems (Miller et all, 1990) and is suitable for real time and on-line practical applications. The CMAC is appropriate for real time and on-line applications because of the properties above (Comoglia, 1991).

The CMAC has a practical hardware realization using logic cell arrays (Miller et all, 1990).

CHAPTER 3

CMAC BASED LIVER ILLNESS DIAGNOSIS SYSTEM

Today, early diagnosis and correct assessment of many diseases have great importance in terms of disease treatment. Therefore, diagnostic and classification process of a disease to be made by using today's technology and medical data would present many uses. In this paper, we have considered the diagnosis of liver diseases such as Hepatitis B, Hepatitis C, Cirrhosis and the cirrhotic phases.

Liver is vital part of our body (Hopkins, 2008). If the liver does not perform any of its vital missions such as production of bile, regulation of blood levels of amino acids, and production of certain proteins for blood plasma etc. (Darwin, 2008), human being would face a serious health consequences. Therefore, early diagnosis of the liver disease is extremely important. When diagnosing the liver disease, the physician looks at the patient's symptoms first such as irregular sleep, jaundice, and portal hypertension, and conducts a physical examination. In addition, the physician may request a liver biopsy, liver function tests (AST, ALT, biluribin), an ultrasound, or a computerized tomography (CT) scan (Darwin, 2008). Liver function tests can diagnose viral hepatitis and autoimmune liver diseases. An ultrasound scan will show blockage of the bile duct, fatty liver, cirrhosis, and liver tumors (Babe, 2007).

During the sate of information gathering related to the liver disease, we made various inferences (Table 3.1). Here, we also would like to deal with the properties of enzymes used as the liver data while we explain one of these inferences.

The physician may spend very long time for the assessment of the enzyme values during normal diagnostic period while making a decision based on those enzymes.

This study provides a contribution to the medical diagnosis process by shortening the time through the use of an intelligent model and helps the physician to diagnose complex cases which are otherwise difficult to perceive. Neural networks have already proven its effectiveness and popularity for the medical diagnostic processes with different existing applications worldwide. In this study, the CMAC neural network approach using human liver test data composed of liver enzymes has been used to diagnose the liver disease in four classes. Enzmyes used to identify the classes were ALT, AST, PT, ALT/AST, Albumin, Protein, PLT. This classification process of the CMAC model is a supervised classification model.

3.1 Enzymes and their signifiance in the Diagnosis of the Liver Disease

As mentioned in the previous section, the first thing that comes to mind is the liver enzyme numbers in diagnosing the liver disease. These enzymes are briefly described as follows:

AST (a.k.a. SGOT) is normally found in a diversity of tissues including liver, heart, muscle, kidney, and brain. For example, its level in serum rises with heart attacks and with muscle disorders. It is therefore, not a highly specific indicator of liver injury (Nabili, 2007), and (Şentürk et all, 2004).

ALT (a.k.a. SGPT) is normally found in the liver. This is not to say that it is exclusively located in liver, but that is where it is most concentrated. It is released into the bloodstream as the result of liver injury. It therefore serves as a fairly specific indicator of liver status (Nabili, 2007), and (Şentürk et all, 2004).

GGT (Gamma-Glutamyl Transpeptidase) is mainly kidney, liver, and pancreas original enzyme. GGT activity rises during all type of liver diseases (Centro, 2008), and (Moseley, 1995). The GGT test is extremely sensitive and may be elevated due to any type of liver disease or by the use of different drugs, including alcohol, even when liver disease is minimal (Batey, and Geoff, 2004).

Albumin is synthesized in the liver. Measurement of total concentration of serum albumin is useful test of liver cells (Moseley, 1995).

Alkaline Phosphatase (a.k.a. ALAP) is a substance found in abundance in the liver and bones. When this enzyme is high, Lecuine aminopeptidase enzyme is checked; if this one is also high, then the damage is said to exist in the liver (Moseley, 1995).

Leucine Aminopeptidase (a.k.a. LAP) is also called a protein that is normally found in liver cells. LAP is released into the blood when your liver cells are damaged. Drugs or infections such as hepatitis can damage liver cells (Alexander, 2007).

Now, we can infer the liver state of a person with regard to the blood levels of these enzymes. First, let us deal with the normal levels of these enzymes (Jaeger, and Hedegaard, 2002), and (Ghange, and Raste, 2004).

AST: Normal Adult Range: 0 - 42 U/L

ALT: Normal Adult Range: 0 - 48 U/L

ALAP: Normal Adult Range: 20 - 125 U/L

GGT: Normal Adult Female Range: 0 - 45 U/L
Normal: Adult Male Range: 0 - 65 U/L

LAP: Normal Adult Range:28-42 U/L

Table 3.1 The enzymes used to diagnose the liver disease.

	Cirrhosis	**Acute Hepatitis (alcohol or drug related)**	**Hepatitis C**	**Non-patient**
ALT	Normal	Increase more than 20 times of the normal	Normal	Higher, Normal
AST	Higher	Increase more than 20 times of the normal	Light or Moderately increased	Higher, Normal
ALP	Icreased up to 3 times of the normal	Light or Moderately increased	Higher	Higher, Normal
LAP	Higher	Higher	Higher	Normal
GGT	Higher, Normal	Higher, Normal	Higher, Normal	Normal

3.2 Pattern Collection

One of the most significant problems of medical diagnosis is the subjectivity of the specialist and the data. Various medical data can be applied to the CMAC models:

1. Electro physical signals like EEG, EKG,

2. Medical Imaging like tomography, ultrasonography or MR,

3. Indicators of disease or tests like blood pressure, blood sugar or cholesterin.

In this study, we have used the data of the published research carried out by Pehlivan and his collaborators (Pehlivan et all., 2008). Hematological, radiological, serological, and biochemical examinations have been carried out on the patients with

the risk of having hemorrhage, and an additional liver biopsy has also been done on the patients with no risk of having hemorrhage, all of whom were included in this study. These results have been considered to diagnose (Comoglio, 1991). The enzyme values obtained as the result of these tests have been used as the liver data as well. Table 3.2 shows the real clinical data used in this thesis in which seven distinct enzyme values have been used for diagnosing the liver disease. These are ALT, AST, PT, ALT/AST, Albumin, Protein, and PLT.

In this study, the samples have been collected from twenty eight patients. Each data set representing one patient consists of approximately eight different attributes. Twenty four of those data have been used for training, and four of them for testing. We have determined four different classes in the liver disease as Hepatitis B (HBV), Hepatitis C (HCV), Cirrhosis A , and Cirrhosis B and C. For each disease, there are six different data collection occurrence.

Table 3.2 Original value of liver enzymes.

İSİM	SOYADI	CİNS	YAŞ	PT	AST	ALT	AST/ALT	TİP	ALB	TPROT	PLT
ŞENEL	ERTÜRKMEN	K	52	12,2	43	155	0,277419	1	3,6	6,7	154000
ABDURRAHMAN	ÖZDEMİR	E	47	13	74	60	1,233333	1	4,5	8,5	221000
ZÜBEYDE	KILIÇ	K	34	13	97	209	0,464115	1	4,4	7,9	244000
AYŞE	HABEŞOĞLU	K	34	12,8	48	69	0,695652	1	3,4	7,6	195000
AYNUR	TURGUT	K	24	12,6	83	172	0,482558	1	3,7	7,1	217000
YAKUP	ÇOLAK	E	24	12,6	74	174	0,425287	1	4,3	8,3	208000
CELAL	SENCEM	E	18	11,6	60	112	0,535714	2	4,7	8,6	251000
HASAN	YAPRAK	E	50	14,6	71	163	0,435583	2	4,3	8,4	117000
FİRDEVS	POLAT	K	47	12,3	28	31	0,903226	2	3,3		216000
ALİ	ÜSTEK	E	67	12,3	32	36	0,888889	2	4,3	8	262000
MEHMET	ÖZYURT	E	49	15,4	116	174	0,666667	2	3,5	8,5	93000
CEVHER	OĞUZ	E	62	23,3	116	134	0,865672	2	4,3	7,4	190000
AHMET	ALTIOK	E	45	16,2	26	53	0,49	3	3,5	7,5	46000
AYŞE	KELEŞ	K	60	17,2	41	15	2,73	3	2,2	6,1	57000
ASİYE	TURAN	K	60	18	37	49	0,75	3	3,2	5,8	120000
DAVUT	AĞBAĞ	E	55	15,9	59	85	0,69	3	3,4	8	75000
GÜLSÜM	BİNGÖL	K	60	16	146	87	1,67	3	2,8	5,6	103000
FİKRET	YARIMAĞA	E	70	16	68	56	1,21	3	3,5	6	64000
NEZAKET	PARLAK	K	33	19	60	32	1,88	4	2,1	5,2	70000
KADİR	YILMAZ	E	55	25	86	51	1,69	4	1,3	5,5	132000
ÜNAL	ÇOŞKUNER	E	55	19,4	57	43	1,30	4	2,6	6,8	101000
ARİF	TORİN	E	60	17,4	295	154	1,90	4	2,1	5,1	177000
HALİL	TAŞ	E	65	29	70	31	2,26	4	1,5	3,6	71000
CUMA	KARAKAYA	E	79	18	56	18	3,11	4	2,3	5,6	79000

Each data has been normalized according to the following formula:

$$normalized\_value = (current\_value - (min\_value - 1)) / ((max\_value - min\_value) + 2) \quad (3.1)$$

According to Eq.3.1, the entire range of the liver data is normalized to vary between 0 and 1, and thereafter the normalized data is used to train and test the CMAC artificial neural network. Figure 3.1 shows how the normalization process is performed

to find the normalized equivalents of the original data for the AST enzyme in this example.

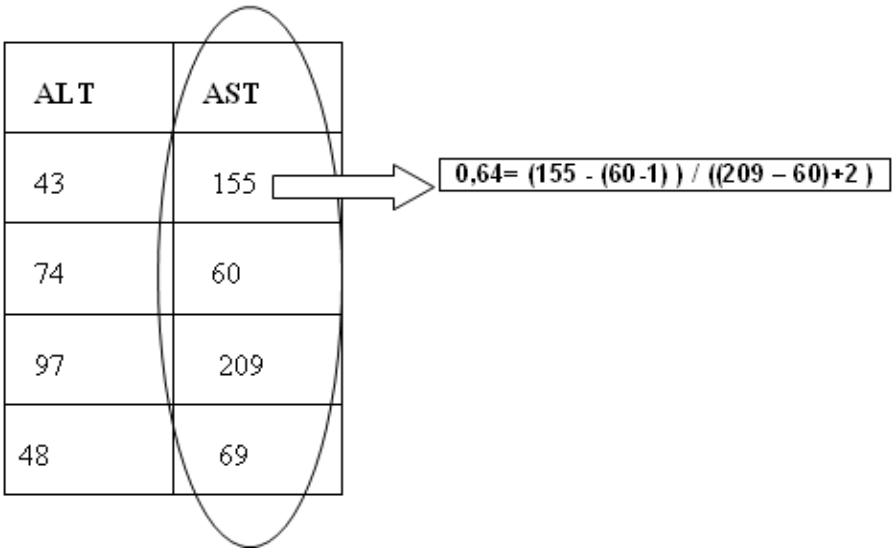

Figure 3.1 An example of normalization process with Eq.(3.1).

3.3 CMAC Liver Diagnosis System

In the previous section, we have mentioned the enzymes used to make a decision for the liver disease. In this section, we are going to mention the liver disease considered in this study with respect to the enzyme values.

In this study, we have succesfully completed the classification of four different diseases toward Hepatitis B, Hepatitis C, Cirrhosis (Phase A), and Cirrhosis (Phases B & C). We have given a brief explanation about these diseases in Chapter 1, we now would like to give an information about the phases of cirrhosis. In this classification process a table of criterion called the Child-Pugh is used. The Child-Pugh score (sometimes the Child-Turcotte-Pugh score) is used to assess the prognosis of chronic liver disease, mainly cirrhosis. Although it was originally used to predict mortality during surgery, it is now used to determine the prognosis, as well as the required strength of treatment and the necessity of liver transplantation (Child, 1964), and (Pugh

et all, 1973). Table 3.3 shows Modified Child-Pugh classification of severity of liver disease according to the degree of ascites, the plasma concentrations of bilirubin and albumin, the prothrombin time, and the degree of encephalopathy (Pugh et all, 1973), and (Lucey et all, 1997).

Table 3.3 Child-Pugh Classification of Severity of Liver Disease.

Parameter	Points assigned		
	1	2	3
Ascites	Absent	Slight	Moderate
Bilirrubin, mg/dL	</= 2	2-3	>3
Albumin, g/dL	>3.5	2.8-3.5	<2.8
Prothrombin time * Seconds over control * INR	 1-3 <1.8	 4-6 1.8-2.3	 >6 >2.3
Encephalopathy	None	Grade 1-2	Grade 3-4

Table 3.4 shows Modified Child Pugh scoring table. A total score of 5-6 is considered grade A (well-compensated disease); 7-9 is grade B (significant functional compromise); and 10-15 is grade C (descompensated disease). These grades correlate with one- and two-year patient survival (Pugh et all, 1973), and (Lucey, 1997).

Table 3.4 Modified Child-Pugh scoring table.

Grade	Points	One-year patient survival (%)	Two-year patient survival (%)
A: well-compensated disease	5-6	100	85
B: significant functional compromise	7-9	80	60
C: descompensated disease	10-15	45	35

Table 3.5 shows the values of cirrhosis data used in this thesis according to the Child-Pugh scoring table.

Table 3.5 The data for the patients with cirrhosis

PT	***AST***	***ALT***	***AST/ALT***	***Albumin***	***Protein***	***Platelet(Trombosit)***	***Child***	***Skor***
15	82	91	0,90	3,6	7	122000	A	5
16,2	26	53	0,49	3,5	7,5	46000	A	5
16	59	55	1,07	3,5	9,7	144000	A	5
14	15	22	0,68	2,5	6,3	98000	A	6
14	62	53	1,17	3,1	7,6	164000	A	6
13	62	106	0,58	3,5	6,8	149000	A	6
16	24	30	0,80	4,4	7,4	172000	A	6
17,2	41	15	2,73	2,2	6,1	57000	A	6
14,4	180	178	1,01	2,9	5,9	62000	B	8
15	69	73	0,95	2,4	6,9	148000	B	8
17,9	18	16	1,13	3	5,9	177000	B	8
17	133	85	1,56	2,6	8,9	101000	B	8
13	49	28	1,75	2,6	5	143000	B	9
16	52	26	2,00	2,7	5,3	180000	B	9
24	23	17	1,35	2,2	5,6	76000	C	10
16,1	37	23	1,61	3	6	89000	C	10
18	97	65	1,49	2,5	5,4	116000	C	10
19	33	36	0,92	2,5	6	65000	C	10

In this study, the CMAC artificial neural network ANN with 24-input vector and 5-output vector has been used. The input vectors can be called training data and the output vectors can be called Hepatitis B, Hepatitis C, Cirrhosis A, Cirrhosis B and C and other types. The class that we call other type can either be any unclassified disease associated with the liver or the liver data with no liver disease. Figure 3.2 shows the CMAC artificial neural network with 24 inputs and 5 outputs.

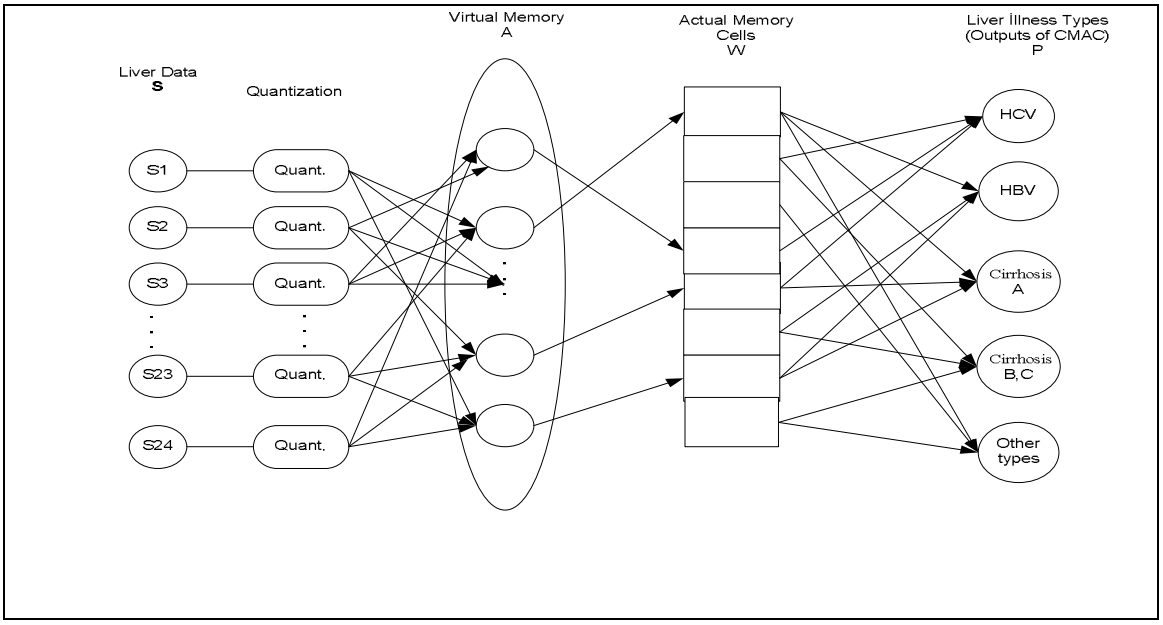

Figure 3.2 The CMAC ANN model used to diagnose the liver disease.

3.3.1 Training mode

In the training mode, the normalized enzyme data are used to train the CMAC artificial neural network. These data perform the mapping process first between quantization and memory locations to start with the network training after being loaded into the CMAC ANN. Later, the output vector is formed by summing the weights in the physical addresses so that the training process gets done. The recognition is decided upon the similarity process which seeks similarity between the output vector of the test data and the training data after the test data has been gone through the similar process as the training data.

3.3.2 Quantization mode

As shown in Figure 3.2, the input signals are first reduced by the quantization process to produce a quantization level output. The quantization output can be calculated in MATLAB as follows (Dunphy, 1993).

```
function [ quantizedInput ] = QuantizeInput( realInput )
global quantization;
[row col]=size(realInput);
%fprintf('Quantization:%d',quantization);
qInput = zeros(col,1);
for k=1:col
   qInput(k)=uint8(floor((realInput(k)-0.00000000000001)*quantization));
   %fprintf('[%d]',qInput(k));
end
%fprintf('\n');
quantizedInput=qInput;
```

According to the equation in the code above, the real input can take values between 0 and the quantization level. For example, if the quantization level is 4, the actual input at the end of the quantization process can take the value of 0, 1, 2, or 3. This process can be realized through the formula provided in the code above.

3.3.3 Learning rule

The CMAC ANN model is a supervised learning algorithm (Hung, and Wang, 2004). According to this algorithm, sample inputs and desired outputs related with the problem are provided to the system. This algorithm uses delta rule during the training as we have mentioned earlier. The CMAC is capable of fast learning because of this learning rule (Moody, 1989).

3.3.4 Learning convergence

To be able to say that the CMAC ANN has the property of learning convergence, then the CMAC should be capable of learning any mapping (Wong, 1992). Miller showed that the CMAC learning rule was LMS (Widrow, and Stearns, 1984), (Miller, 1989), and (Miller et all, 1990), and LMS rule was not a satisfactory one in terms of a global learning convergence as it only guaranteed local minima (Wong, 1992). If the CMAC had a big enough memory for the mapping between the virtual memory and the physical memory, the research has shown that there is no need for hashing (Wong, 1992). It is proven that the CMAC is capable of learning any mapping, (Wong, 1992). As a result, the CMAC NN satisfies the learning convergence, i.e., it guarantees it (Wong, 1992), (Chiang, 1995), and (Hung, and Yang, 2007).

3.4 Diagnosis Mode

The trained CMAC NN is now ready for the recognition process. The totally different data from the data of the training, which goes through the same normalization process and is called test data, are inputted to the network for the recognition process. The operations of the CMAC ANN will be the same as the training mode when the test data is inputted to the diagnostic system. But in diagnosis mode, the weights of the same excited memory addresses of each memory layer are summed up and each layer has one output value. If the input signals are the same as the training patterns, they will excite the same memory addresses (Hung, and Yang, 2007). So, the output of CMAC ANN can be HBV, HCV, Cirrhosis A, Cirrhosis B and C; Otherwise, the output of CMAC ANN will be called other types in the program.

3.4.1 Train and Diagnosis algorithm

In this section, the CMAC algorithm is described as based on the configuration in Figure 3.2:

Step 1 Build configuration of CMAC liver illness diagnostic system. It

includes 24-input liver data and 5 output nodes.

Step 2 Normalize, load and input the *training* data, through quantization, memory addressing, and the weights of the summation of excited memory addresses to produce the output nodes.

Step 3 Calculate the difference between actual output and desired output to find the weights, which minimize the error as based on the LMS rule.

Step 4 Training is finished. Save the memory weights.

Step 5 Normalize, load and input the *testing* data, through quantization, memory addressing, and summation of the weights of the excited memory addresses to produce the outputs. (If the input signals are the same as the training patterns, they will excite the same memory addresses).

Step 6 Output the testing result.

We initially studied a code (Dunphy, 1993). We have developed our own code as based on our own needs. The algorithm mentioned above has been implemented by MATLAB programming language with the codes shown below. This MATLAB program is composed of seven separate sections. These sections are CMAC_run, Initialize CMAC, QuantizeInput, FromInterconnection, Train, Test and ComputeOutput respectively. We will deal with each section separately and explain the significant details.

In the first section, the train and test data are loaded, and the digital values of the quantization, maximum iteration, Learning rate (β), desired error are expressed as a matrix of the size 1*5 with a name of Prm. Later on, the desired output is reached by invoking initialization, training, quantization, and testing functions respectively.

```
clear all; clc;
global n_inputs;    global pass;            global maxIteration;
global data;        global input;           global error;
global desired;     global totalError;      global maxError;
global desiredError;global test;            global inputCount;
global output;
fprintf('Loading Files...\n');
load traindata.txt;
load testdata.txt;
fprintf('Initializating CMAC...\n');
Prm=[4 2 200 0.1 0.1];%[quantization width maxIteration learningRate targetError]
timeInitStart=clock;
initializeCMAC(Prm(1),Prm(2),Prm(3),Prm(4),traindata,testdata,Prm(5));

timeInitFinish=clock;
fprintf('Initialization is Completed.\nStarting Training\n');
timeTrainingStart=clock;
maxError=1.0;
while (maxError>=desiredError) &&(pass<maxIteration)     % for each pass */
       totalError=0.0;
       maxError=0.0;
       for i=0:n_inputs-1                          % for each possible input */
          input = data(i+1,1:inputCount);
          input = QuantizeInput(input);
          desired = data (i+1,inputCount+1);
          Compute_Output();
          Train();
          fprintf('Input:');
          for sc=1:inputCount
             fprintf('[%d]',input(sc));
          end
          fprintf('\tTarget:%d\tActual:%f\t\tError:%f\n',desired,abs(output),error);
         totalError=totalError+abs(error);
         maxError=max(abs(error),maxError);
       end


fprintf('PASS=%d   \tMaximum Error=%g\tAverage Error=%f\n', pass,maxError,(totalError/n_inputs));
```

```
fprintf('_____
_\n');
      pass=pass+1;
      if pass>maxIteration
        break;
      end
end
timeTrainingFinish=clock;
fprintf('\nTraining is Completed !\nStarting Testing....\n');
timeTestStart=clock;
TestCMAC(test);
timeTestFinish=clock;
initDuration = etime(timeInitFinish,timeInitStart);
trainingDuration = etime(timeTrainingFinish,timeTrainingStart);
testDuration = etime(timeTestFinish,timeTestStart);
fprintf('\nInitialization Duration:%f\n',initDuration);
fprintf('Training Duration:%f\n',trainingDuration);
fprintf('Test Duration:%f\n',testDuration);
fprintf('Training Step:%d\n',pass-1);
fprintf('%d\t%d\t%f\t%f\t%d\t%f\t%f\t%f\n',Prm(1),Prm(2),Prm(4),Prm(5),pass-
1,initDuration,trainingDuration,testDuration);
```

In this section, initalization process is started by receiving the values inside Prm[.] matrix. Firstly, the size of the traning data is computed, and then, the input dimension is determined, and lastly, the possible number of input vectors are determined. The other most significant part of this section is to input the maximum iteration number to the program. Then, FromInterCon() function, which makes the intermemory mapping operations, is invoked.

```
function InitializeCMAC( quant,w,maxIt,beta,trainData,testData,targetError)
global n_inputs;    global pass;            global n_sensors;
global inputCount;  global quantization;    global width;
global maxNeurons;  global conn;            global maxIteration;
global data;        global weight;          global learningRate;
global desiredError;global test;            global numberOfNeurons;
    data=trainData
   test =testData
   recordCount=size(data);
   inputCount=recordCount(2)-1 ;% input dimensions */
   n_inputs=recordCount(1);  % number of possible input vectors */
   quantization=quant; % input quantization per dimension */
   width=w ;% width of input sensors */
   maxNeurons=500000;
   n_sensors = (quantization + width - 1) ^ inputCount;
   conn = zeros(quantization,inputCount,maxNeurons+5);
   numberOfNeurons=FormInterConnection()
   n_sensors=numberOfNeurons * inputCount;
   weight = zeros(numberOfNeurons+1,1);
   pass=1;
   desiredError=targetError; % maximum error for any input */
   maxIteration = maxIt;
   learningRate = beta;
   %fprintf('Sensor Count:%d\n',n_sensors);
   fprintf('Training Data Row Count:%d, Col Count:%d\n',n_inputs,inputCount);
   fprintf('Quantization:%d\n',quantization);
   fprintf('Target Min Error:%f\n',desiredError)
```

This section does the mapping operations between the virtual memory and the physical memory.

```
function [neuronCount] =FormInterConnection( )
global maxNeurons;      global quantization;    global width;
global conn;            global n_sensors;       global inputCount;
%global numberOfNeurons;
```

```
numberOfNeurons=0;
k1 = (quantization + width - 1); % intermediate calculation */
      for i=0:width-1
        for j=0:n_sensors-1
          found=1;
            m=1;
          for k=0:inputCount-1
            mod0=floor(j/m);
            mod1=mod(mod0,k1);
            mod1=floor(mod1);
            mod2=mod(mod1,width);
            if  mod2 ~=i          %if((((j/m)%k1)%width)!=i){
              found=0;
              break;
            end
            m=m*k1;
          end
          if found==1
                m=1;
                for k=0:inputCount-1
                  n=mod(floor(j/m),k1);
                  for p=max(0,n-width+1): min(n,(quantization-1))
                    col=conn(p+1,k+1,maxNeurons+1)+1;
                    conn(p+1,+k+1,col) =numberOfNeurons;

                    num=conn(p+1,k+1,maxNeurons+1);
                    conn(p+1,k+1,maxNeurons+1)=num+1;
                  end
                  m=m*k1;
                end
                numberOfNeurons=numberOfNeurons+1;
                  if numberOfNeurons > maxNeurons
                    fprintf('cmac: error, maximum number of gates exceeded');
                  end
              end
        end
      end
  neuronCount=numberOfNeurons;
  fprintf('Mapping Completed !\n');
```

This function forms an output for all the given input values by being invoked after the initialization process.

```
function Compute_Output( )
global output;      global activatedNeurons;    global input;
global weight;      global neurons;             global inputCount;
global maxNeurons;  global currentNeuronCount;  global conn;
global numberOfNeurons;
  activatedNeurons(numberOfNeurons+1)=0;
      output=0;
      neurons = zeros(numberOfNeurons,1);
        for u=0:inputCount-1                              % for all inputCounts */
            inp=input(u+1);
            currentNeuronCount=conn(inp+1,u+1,maxNeurons+1);
            for v=0:currentNeuronCount-1                  %increment all Neurons in list */
                g=conn(inp+1,u+1,v+1);
                neurons(g+1)=neurons(g+1)+1;
                if ((u+1)==inputCount) && (neurons(g+1)==inputCount)      % if activated */
                    pu=  activatedNeurons(numberOfNeurons+1);
                    activatedNeurons(numberOfNeurons+1)=pu+1; % generate list of activ'd Neurons */
                    activatedNeurons(pu +1) =g;
                    output=output + weight(g+1);
                end
            end
          end
```

Train function is essentially the weight update process in which firstly compares the actual output with the desired output is compared, and then the weights of the active neurons if a difference exists between the actual and desired outputs are updated with the predetermined learning rate according to the LMS (Eq. 2.1). This is basically no different from updating the weights of the active neurons.

```
function  Train( )
global inputCount;      global numberOfNeurons; global output;
global activatedNeurons;global input;          global weight;
global learningRate;    global error;          global desired;
```

```
error=desired-output;
for E=0:activatedNeurons(numberOfNeurons+1)-1
   % using list of activated neurons */
    ind=activatedNeurons(E+1)+1;
    agirlik=weight(ind);
    hesap=agirlik+ (learningRate*error);
    weight(ind)=hesap;
end
```

This is a lastly invoked function. It determines the size of the test data and assigns the inputs in the data file to a matrix. Later on, it quantizes the inputs in this matrix one by one, it forms an output by invoking Compute-Output function. Lastly, it classifies according to the output value.

```
function TestCMAC( test )
global input;
global output;
[testRowCount,testColCount] = size(test);
for i=1:testRowCount
   input = test(i,1:testColCount-1);
   fprintf('Test %d\nReal Input:\t',i);
   for j=1:testColCount-1
      fprintf('[%f]',input(j));
   end
   input=QuantizeInput(input);
   fprintf('\nQuantized:\t');
   for j=1:testColCount-1

      fprintf('[%d]',input(j));
   end
   Compute_Output();
   fprintf('\nOutput:%f\t',output);
   intOutput = uint8(output);
   switch intOutput
      case 1
         fprintf('(HVB)');
```

```
        case 2
            fprintf('(HVC)');
        case 3
            fprintf('(child-A)');
        case 4
            fprintf('(child-BC)');
        otherwise
            fprintf('(Other Types)');
    end
    fprintf('\n_____\n');
end
```

3.5. Case study and discussion

To demonstrate the effectiveness of the proposed algorithm, we have sieved to seven sets of data for each liver illness type. Six sets are utilized as the training pattern and the last one is the test data. All the data are listed in Table 3.6 and the bold typed rows represent the test data.

Table 3.6 Liver test and train data·

Patient no	Normalized liver data								Illness Type
	Age	ALT	AST	ALT/AST	Albumin	Protein	Platelet	PT	
1	0,56	0,05	0,07	0,72	0,21	0,61	0,59	0,42	HBV
2	0,48	0,1	0,18	0,22	0,4	0,78	0,86	0,68	HBV
3	0,27	0,1	0,27	0,99	0,25	0,76	0,77	0,77	HBV
4	0,27	0,09	0,08	0,27	0,29	0,57	0,72	0,58	HBV
5	0,11	0,07	0,21	0,8	0,25	0,63	0,65	0,66	HBV
6	0,11	0,07	0,18	0,81	0,24	0,74	0,83	0,63	HBV
7	0,02	0,02	0,13	0,49	0,26	0,81	0,87	0,8	HCV
8	0,52	0,18	0,17	0,76	0,24	0,74	0,84	0,28	HCV
9	0,48	0,06	0,01	0,07	0,34	0,56	0,78	0,66	HCV
10	0,79	0,06	0,03	0,1	0,33	0,74	0,78	0,84	HCV
11	0,51	0,22	0,34	0,81	0,29	0,59	0,86	0,18	HCV
12	0,71	0,64	0,34	0,61	0,33	0,74	0,7	0,56	HCV
13	0,44	0,27	0	0,19	0,25	0,59	0,71	0	Child-A
14	0,68	0,32	0,06	0,01	0,72	0,35	0,51	0,04	Child-A
15	0,68	0,36	0,04	0,17	0,3	0,54	0,46	0,29	Child-A
16	0,6	0,25	0,13	0,35	0,29	0,57	0,78	0,11	Child-A
17	0,68	0,26	0,45	0,36	0,5	0,46	0,43	0,22	Child-A
18	0,84	0,26	0,16	0,2	0,4	0,59	0,49	0,07	Child-A
19	0,25	0,41	0,13	0,08	0,54	0,33	0,38	0,09	Child-BC
20	0,6	0,73	0,23	0,18	0,5	0,19	0,42	0,33	Child-BC
21	0,6	0,44	0,12	0,13	0,42	0,43	0,61	0,21	Child-BC
22	0,68	0,33	1	0,71	0,54	0,33	0,36	0,51	Child-BC
23	0,76	0,95	0,17	0,07	0,62	0,22	0,14	0,1	Child-BC
24	0,98	0,36	0,11	0,01	0,79	0,37	0,43	0,13	Child-BC
25	**0,12**	**0,07**	**0,2**	**0,8**	**0,25**	**0,62**	**0,65**	**0,65**	HBV
26	**0,51**	**0,22**	**0,34**	**0,81**	**0,29**	**0,59**	**0,86**	**0,18**	HCV
27	**0,68**	**0,36**	**0,04**	**0,17**	**0,3**	**0,54**	**0,46**	**0,29**	Child-A
28	**0,6**	**0,43**	**0,11**	**0,13**	**0,42**	**0,43**	**0,61**	**0,21**	Child-BC

Figure 3.3 gives us the results of the CMAC NN program which we have formed with a predetermined quantization, width, learning rate, and the desired error values. Table 3.7 shows the obtained results when we give various values to the variables.

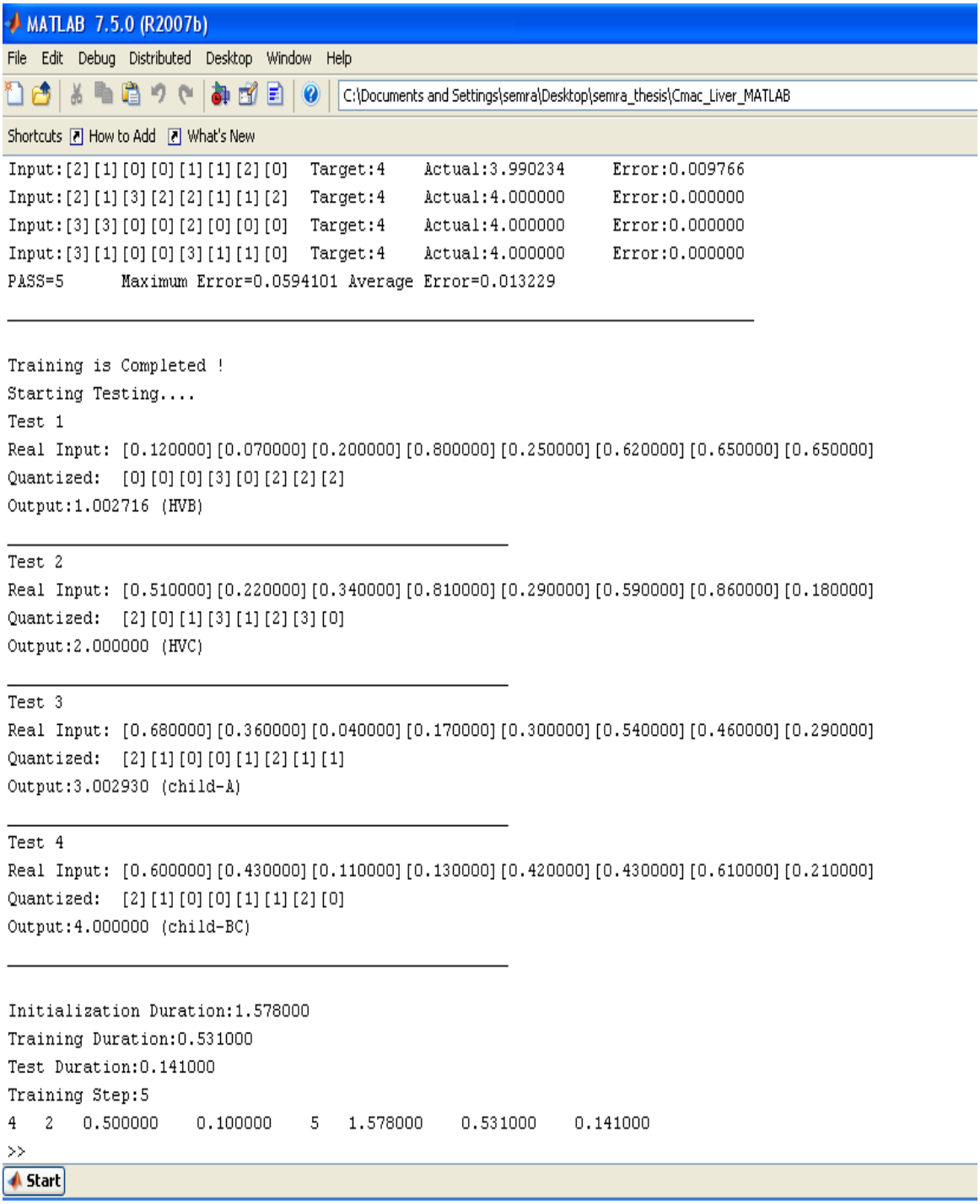

Figure 3.3 Output of the MATLAB program.

Table 3.7 Performance table of the MATLAB CMAC program.

Q	W	Beta	Desired Error	Init Time	Learning T.	Test Time	Step
3	2	0,1	0,1	0.34400	3.765.000	0.016000	199
3	2	0,1	0,01	0.312000	3.719.000	0.015000	199
3	2	0,1	0,001	0.328000	3.609.000	0.016000	199
3	2	0,1	0,0001	0.313000	3.625.000	0.016000	199
3	2	0,1	0,00001	0.312000	3.735.000	0.015000	199
3	2	0,1	0,000001	0.328000	4.000.000	0.016000	199
3	2	0,4	0,1	0.328000	3.625.000	0.016000	199
3	2	0,4	0,01	0.328000	3.625.000	0.016001	199
3	2	0,4	0,001	0.328000	3.578.000	0.016000	199
3	2	0,4	0,0001	0.328000	3.750.000	0.016000	199
3	2	0,4	0,00001	0.329000	3.593.000	0.016000	199
3	2	0,4	0,000001	0.313000	3.610.000	0.015000	199
4	2	0,4	0,1	1.375.000	0.469000	0.016000	6
4	2	0,4	0,01	1.344.000	0.656000	0.031000	9
4	2	0,4	0,001	1.390.000	0.891000	0.031000	13
4	2	0,4	0,0001	1.391.000	1.187.000	0.031000	17
4	2	0,4	0,00001	1.359.000	1.407.000	0.031000	20
4	2	0,4	0,000001	1.359.000	1.657.000	0.031000	24
4	3	0,4	0,1	13.453.000	0.750000	0.032000	16
4	3	0,4	0,01	13.453.000	1.234.000	0.032000	27
4	3	0,4	0,001	13.656.000	1.750.000	0.031000	37
4	3	0,4	0,0001	13.281.000	2.141.000	0.031000	47
4	3	0,4	0,00001	13.406.000	2.750.000	0.031000	58
4	3	0,4	0,000001	13.609.000	3.360.000	0.157000	68
5	3	0,4	0,1	46.140.000	2.172.000	0.063000	16
5	3	0,4	0,01	45.718.000	4.985.000	0.047000	38
5	3	0,4	0,001	45.984.000	7.735.000	0.046000	60
5	3	0,4	0,0001	45.844.000	11.188.000	0.062000	82
5	3	0,4	0,00001	45.938.000	13.907.000	0.062000	104
5	3	0,4	0,000001	45.843.000	16.437.000	0.047000	126
4	2	0,6	0,1	3.015.000	0.719000	0.031000	6
4	2	0,6	0,01	2.984.000	1.063.000	0.047000	9
4	2	0,6	0,001	3.016.000	1.532.000	0.046000	13
4	2	0,6	0,0001	3.032.000	1.875.000	0.046000	16
4	2	0,6	0,00001	3.016.000	2.265.000	0.031000	19
4	2	0,6	0,000001	2.985.000	2.750.000	0.047000	23
4	2	0,5	0,1	1.500.000	0.406000	0.157000	5
4	2	0,5	0,01	1.344.000	0.593000	0.016000	8
4	2	0,5	0,001	1.344.000	0.719000	0.015000	11
4	2	0,5	0,0001	1.344.000	0.969000	0.031000	14
4	2	0,5	0,00001	1.359.000	1.250.000	0.015000	17
4	2	0,5	0,000001	1.359.000	1.360.000	0.031000	20

4	2	0,3	0,1	1.329.000	0.484000	0.016000	11
4	2	0,3	0,01	1.344.000	0.938000	0.031000	14
4	2	0,3	0,001	1.359.000	1.250.000	0.031000	18
4	2	0,3	0,0001	1.359.000	1.641.000	0.031000	24
4	2	0,3	0,00001	1.343.000	2.047.000	0.031000	29
4	2	0,3	0,000001	1.375.000	2.329.000	0.015000	34
4	2	0,8	0,1	1.375.000	0.766000	0.031000	11
4	2	0,8	0,01	1.375.000	1.328.000	0.015000	19
4	2	0,8	0,001	1.375.000	2.156.000	0.031000	27
4	2	0,8	0,0001	1.375.000	2.532.000	0.031000	36
4	2	0,8	0,00001	1.344.000	3.000.000	0.015000	44
4	2	0,8	0,000001	1.375.000	3.609.000	0.016000	53

According to Table 3.7, the variables of *quantization=4, width=2, learning rate=0,5* have shown us that the program runs with the best performance overall.

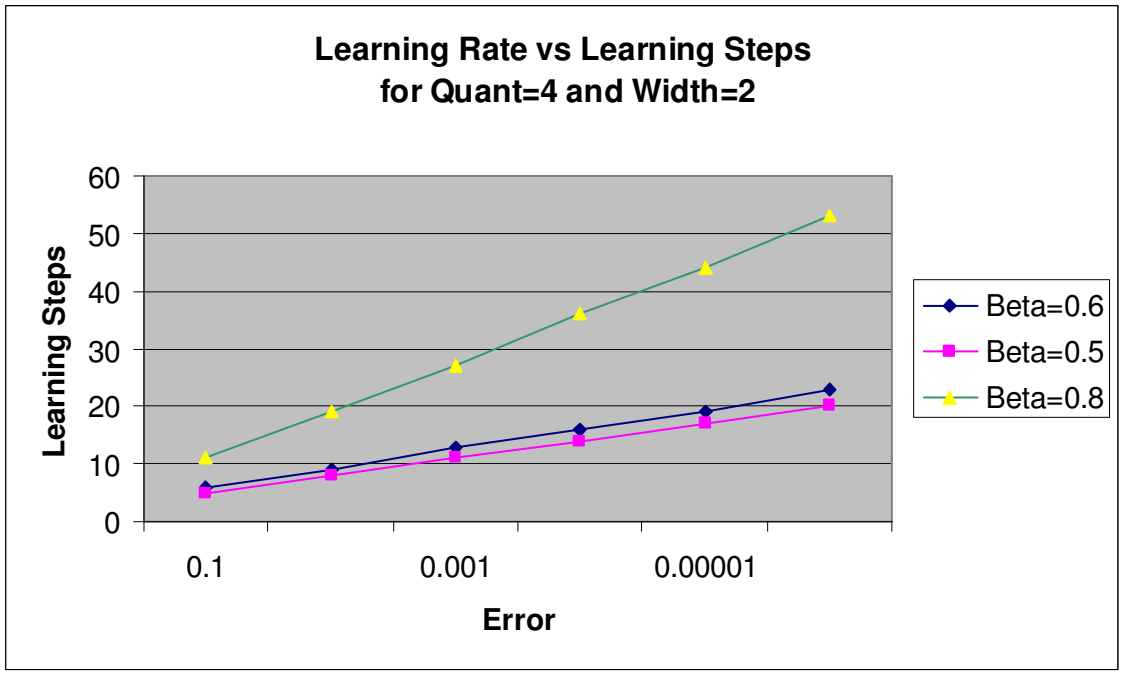

Figure 3.4 Learning rate vs. Learning steps.

Figure 3.4 shows that, as the learning rate β is decreased, it takes shorter steps for the algorithm to classify the input states accurately. That can also be justified from the resulting table data in Table 3.7.

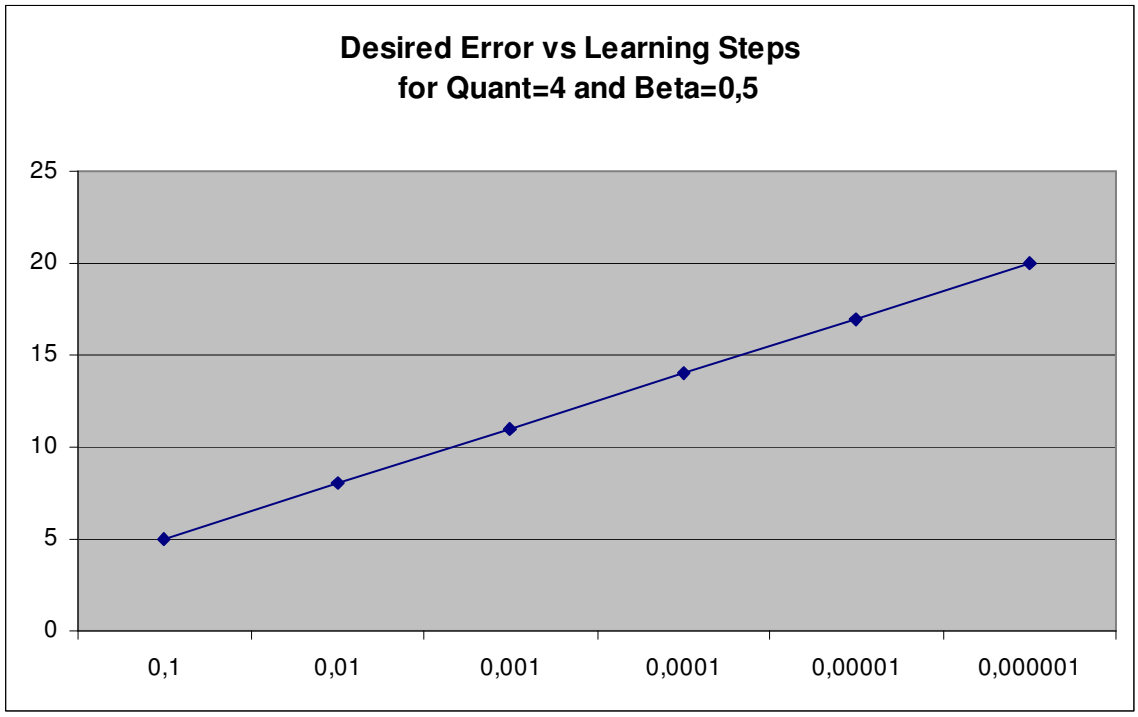

Figure 3.5 Desired error vs. Learning steps.

Figure 3.5 indicates that the learning steps for the algorithm that learns to classify the data correctly will increase linearly for all the desired errors.

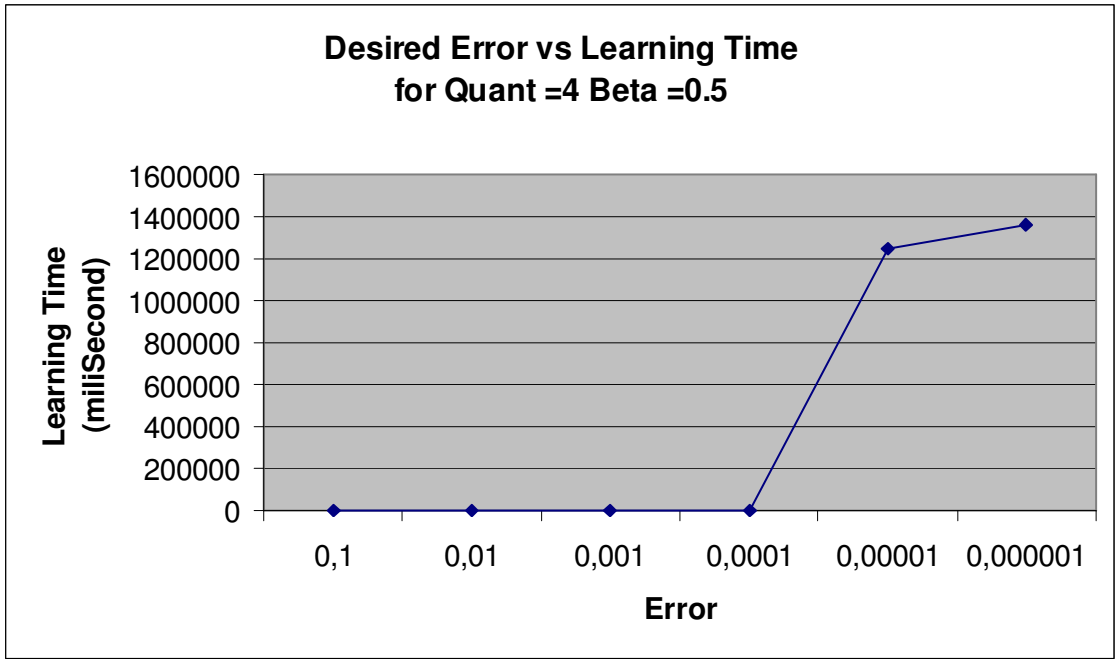

Figure 3.6 Desired error vs. Learning time.

Figure 3.6 shows the relationship between the desired error and the learning time when the quantization is 4, and the learning rate is 0,5. According to the figure, there is no any recordable variation in the learning time until the desired error becomes 0.0001. Nonetheless, the learning time increases sharply at the desired error which is much smaller than the value of 0.0001 as shown in the figure.

CHAPTER 4

CONCLUSION

In this chapter we will summarize the major findings of our study. As the use of artificial neural networks in various recognition processes is widespread, the applications of the CMAC ANN to such recognition processes has been known for a long time. Melancholia Diagnosis based on CMAC Neural Network Approach (Hung, and Yang, 2007), and CMAC based neural networks detection for drinking water quality (Bucak, 2008) can be given as the two recent applications in this field.

In this thesis, we provide an alternative way to medical diagnosis. The physician may spend very long time for the assessment of the enzyme numbers during normal diagnostic period while making a decision based on those enzymes. This study provides a contribution to the medical diagnosis process by shortening the time through the use of an intelligent model and helps the physician to diagnose complex cases which are otherwise difficult to perceive. Physicians make a decision according to enzyme values in normal diagnosis stage of this method .

In addition, with the learning ability of the CMAC neural network models, adaptability to various problems, minimum data requirements and minimum processing time period help modeling problems efficiently.

A system to recognize all types or phases of the liver disease in the future can be developed by using the data and the method in this study. The future research may present many advantages in the recognition of the process and the performance of the algorithm by increasing both the number of training and the testing data. Other enzymes not used in this study can be included to the research to refine and diversify the results. A comparison can be made between the current algorithm and a future extraction method to be used with the liver data in this study. Additionally, a fuzzy neural network

application can serve to explain and illustrate the unknown transitional period between the phases of the liver disease.

REFERENCES

Adams, B., *Liver, Biliary, & Pancreatic Disorders*, 2008, http://www.healthsystem.virginia.edu/uvahealth/adult_liver/virhepov.cfm

Babe, T., *The Liver and its Diseases*, 2007, http://www.bbc.co.uk/dna/h2g2/A134920

B. Karlık and I. O. Bucak, "Hazardous Odor Recognition by CMAC Based Neural Networks, *5th International Advanced Technologies Symposium (IATS'09)*, Karabük, Turkey 13-15 May 2009.

B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Englewood Cliffs, NJ: Prentice- Hall, 1984.

Centro, G., *Gama Glutamil Transpeptidaz (GGT),* 1997 http://www.centro.com.tr/testarama.asp?action=view&viewid=27

C. P. Hung, Mang-Hui Wang, "Diagnosis of incipient faults in power transfers using CMAC Neural Network approach", *Electric Power Systems Research*, Vol. 71, pp235-244, 2004.

Chin-Pao Hung, Mang-Hui Wang, Chin-Hsing Cheng and Wen-Lang Lin, "Fault Diagnosis of Steam Turbine- generator Using CMAC Neural Network Approach", *Neural Networks, 2003. Proceedings of the International Joint Conference of IEEE*, 20-24 July 2003, Vol.4, pp.2988-2993, 2003.

Chin-Pao Hung , Shi-Liang Yang, "Melancholia diagnosis based on CMAC neural network approach", *Proceedings of the 8th Conference on 8th WSEAS International Conference on Neural Networks,* 19-21 June, 2007, p.25-30, Vancouver, British Columbia, Canada , 2007.

Child CG, Turcotte JG., "Surgery and portal hypertension. In: The liver and portal hypertension", *Saunders, Philadelphia*, pp.50-64, 1964.

C. P. Hung, Yi-shin, Wei-Ging Liu, "PIC microcontroller based fault diagnosis apparatus design for water circulation system using CMAC neural network approach", *WSEAS Trans. On Information Science & Application*, Vol. 4, Issue 2, pp. 393-399, February, 2007.

C. T. Chiang, *CMAC Addressing Technique Based Learning Structures*, the Faculty of Graudate School the University of Missouri- Colombia, May, 1995.

Daniel R. Alexander, *Health Information,* Department of Internal Medicine, St. Mary's Hospital, Leonardtown, MD. Review provided by VeriMed Healthcare Network, 5, 2007.

Darwin, P., *Liver Disease*, 2008, http://www.umm.edu/liver/liver.htm

Dimitriou, D., 2008, *What is Hepatitis*, 2008, http://www.hepatitis.org.uk/s-crina/whatis-fs.htm

Dupage, K., *Liver Disease Symptoms*, 2007, http://www.healthinfoarticles.com/liver-disease-symptom.html

D. Rumelhart and J. McClelland, Paralel Distributed Processing: Explorations in the Microstructere of Cognition*, Foundations, MA: MIT Pres,* Vol. 1, 1986.

George Burgin, "Using Cerebellar Arithmetic Computers", *AI Expert* , pp. 32-41, June 1992.

Gott, *Liver Cirrhosis*, 2008, http://www.mamashealth.com/stomach/livcir.asp

Hakan Şentürk, Billur Canbakan, İbrahim Hatemi, "Karaciğer Enzim Yüksekliklerine Klinik Yaklaşım", *İ.Ü. Cerrahpaşa Tıp Fakültesi Sürekli Tıp Eğitimi Etkinlikleri, Gastroenterolojide Klinik Yaklaşım Sempozyum Dizisi*, No: 38 , pp.9-13, 2004.

Handelman, D.A., S. H. Lane, and J.J. Gelfand, "Integrating neural networks and knowledge-based systems for intelligent robotic control", *IEEE Control Systems Magazine*, vol. 10, no. 3, pp. 77-87, 1990.

Hopkins, J., *Liver*, 2008, http://en.wikipedia.org/wiki/Liver

Howard Worman, "*The Hepatitis C Council of NSW Inc is a community-based, non-government organisation"*, funded by the NSW Health Department, April 2004. (Adapted with assistance from Prof Bob Batey & Prof Geoff Farrell from an article in PositiveLiving, using additional internet information by (USA)).

Hung-Ching Lu, Ming-Feng Yeh, and Jui-Chi Chang, "CMAC Study with Adaptive Quantization", *2006 IEEE International Conference on Systems, Man, and Cybernetics*, 8-11 October, 2006, pp. 2956-2601, Taipei, Taiwan, 2006.

I.O. Bucak, "CMAC Based Neural Networks Detection for Drinking Water Quality", *Regional Process of the 5th World Water Forum*, Regional Meeting on Water in the Mediterranean Basin, Oct. 2008, TRNC, 2008.

Ji Zhu, Xiao-Dong Zhu, Shi-Xiong Liang, Zi-Yong Xu, Jian-Dong Zhao, Qi- Fang Huang, An-Yu Wang, Long Chen, Xiao-Long Fu, and Guo-Liang Jiang, "Prediction of Radiation Induced Liver Disease Using Artificial Neural Networks", *Jpn J Clin Oncol*, Vol.36, No.12, pp.783–788, 2006.

J. J. Jaeger and H. Hedegaard, *About blood tests*, 2002, http://home3.inet.tele.dk/omni/alttest.htm#sgot

J.Moody, "Fast learning in multi-resolution hierarchies," *Advances in Neural Information Processing Syst.* 1, D.S. Touretzy, Ed. Los Altos, 1989, pp. 29-39, 1989.

J.S. Albus, "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)", *Journal of Dyn. Syst. Meas. Control*, Trans. ASME 97, pp.220-227, 1975.

J.S. Albus, A New Approach to Manipulator Control: *The Cerebellar Model Articulation Controller (CMAC)*, NIST - National Institute of Standards and Technology, 237-240, 1975.

L.G. Kraft and D.P. Campanga, "A comparison between CMAC neural network control and two traditional adaptive control systems", *IEEE Contr. Syst. Mag.*, pp.36-43, Apr. 1990.

Lucey MR, Brown KA, Everson GT, Fung JJ, Gish R, Keeffe EB, Kneteman NM, Lake JR, Martin P, McDiarmid SV, Rakela J, Shiffman ML, So SK, Wiesner RH, "Minimal criteria for placement of adults on the liver transplant waiting list*", Liver Transpl Surg.*, Vol.3, No.6, pp.628-37. November, 1997.

McLaughlin, E., Liver Disease, 2000, http://uimc.discoveryhospital.com/main.php?id=3307

Doebelin, E., *Control System Principles and Design*, John Wiley, New York, 1985.

M. Hormel, *A self- organizing associative memory system for control applications, in Advances in Neural Information Processing Systems 2*, Morgan Kaufmann, 1990.

Michael T Dunphy, *CMAC - Cerebellum Model Articulation Controller*, 1993 http://www.cs.cmu.edu/afs/cs/project/airepository/ai/areas/neural/systems/cmac/cmac.txt

M.S. Ghadge and A.S. Raste, "Leusine Amino Peptidase a beter indicator of disseminated maligmant disease", *Indian Journal of Clinical Biochemistry*, Department of Biochemistry, Tata Memorial Hospital, Parel, pp. 149-151, 2004.

Nabili, S., *Liver Blood Test*, 2008 http://www.medicinenet.com/liver_blood_tests/article.htm

Petska, J., *What is the Largest Gland in the Body?,* 2007, http://www.ehow.com/about_4570848_what-largest-gland-body.html

Pugh RN, Murray-Lyon IM, Dawson JL, Pietroni MC, Williams R., "Transection of the oesophagus for bleeding oesophageal varices", *Br J Surg.* 60(8):646-9. Aug, 1973.

Richard H. Moseley, *Approach to the Patient With Abnormal Liver Chemistries*, Textbook of Gastoenterology, Tadataka Yamada, pp.909-922, JB Lippincott Company, Philadelphia, 1995.

Rick F. Comoglio, *Using a Cerebellar Model Arithmetic Computer Neural Network to Control An Autonomous Underwater Vehicle*, M.S. Thesis Faculty of Florida Atlantic University, 1991.

Semra Icer, Sadik Kara, "Spectral analysing of portal vein Doppler signals in the Cirrhosis patients", *Computers in Biology and Medicine 37*, pp.1303-1307, December, 2006.

Y. Pehlivan, M. Koruk, M. T. Gülşen, C. Savaş, A. Kadayıfçı, "The Relation Between AST, ALT Ratio And Stage of The Disease in Chronic Viral Hepatitis - Importance of AST/ALT in Chronic Hepatitis", *Gaziantep University Medicine Magazine*, Vol. 14, pp.28-31, 2008.

Wong, Y. F., Sideris, A., "Learning convergence in the cerebellar model articulation controller", *IEEE Trans. on Neural Network*, Vol. 3, No. 1, pp.115-121, 1992.

W.Thomas Miller, Filson H. Glanz and L. Gordon Kraft, "CMAC: An Associative Neural Network Alternative to Backpropagation", *IEEE Proceedings*, Vol. 75, pp. 1561-1567, October 1990.

W. T. Miller and Filson H. Glanz, *The University of New Hampshire Implementtion of tthe Cerebellar Model Arithmetic Computer-CMAC*, Robotics Laboratory University of New Hampshire Durham, New Hampshire No.03824, August 31,1994 (last modified July, 1996).

W.T. Miller, B. A. Box, and E. C. Whitney, *Design and implemantation of high speed CMAC neural network using programmable CMOS logic celle arrays*, University of New Hampshire, Rept. No. ECE.IS.90.01, Feb.6, 1990.

W. T. Miller, " Real-time application of neural networks for sensor-based control of robot with vision", *IEEE Trans. Syst*., Man, Cybern., Vol. 19, pp. 825-831, 1989.

W. T. Miller, R. P. Hewes, F. H. Glanz, and L. G. Kraft, "Real- time dynamic control of an industrial manipulator using a neural- network- based learning contreoller", *IEEE Trans. on Robotics and Automation*, Vol. 5, No. 1, pp.1-9, February, 1990.