ARTIFICIAL NEURAL NETWORK FOR GAIT DISORDER CLASSIFICATION

by

Shavkat Kuchimov

B.Sc. in Electrical and Electronic Engineering, Boğaziçi University, 2002

Submitted to the Institute of Biomedical Engineering
in partial fulfillment of the requirements
for the degree of
Master of Science
in
Biomedical Engineering

Boğaziçi University
September 2006

ARTIFICIAL NEURAL NETWORK FOR GAIT DISORDER CLASSIFICATION

APPROVED BY:

Prof. Dr. Mehmed Özkan .....
(Thesis Supervisor)

Prof. Dr. Yener Temelli ……

Assistant Prof. Dr Can Yücesoy ……

DATE OF APPROVAL: 09.09.2006

ACKNOWLEDGEMENTS

I would like to gratefully acknowledge the Institute of Biomedical Engineering at Boğaziçi University, and the Motion Analysis Laboratory at Istanbul University Istanbul Faculty of Medicine, where the research and data acquirements were carried out.

I am deeply grateful to Prof. Dr. M. Özkan for his consultancy during my thesis work. He advised me Motion Analysis Laboratory at Istanbul University Istanbul Faculty of Medicine. He always motivates me whenever I have hard times with research. He made his patience and supervision available at all times. Therefore, it was impossible to progress in this project without his encouragements and great efforts.

I would like to express my gratitude to Prof. Dr. Yener Temelli and Mr. N. Ekin Akalan. They provided me with warm laboratory environment, which is rare in hospitals. They gave me opportunity to easily get data acquirement and were always ready to share their invaluable medical knowledge.

I am indebt to my colleagues: Gülay, Sanem, Sibel and Saadet in Motion Analysis Laboratory. They are my companions who were always ready to help.

I would like to address special thanks to my family. They give me moral help and patience. Although I am far away from my homeland, they make me feel at home with their maternal love.

This project is consequence of efforts and patience of all those great people. I would like to devote this project to all of them.

ABSTRACT

ARTIFICIAL NEURAL NETWORK FOR GAIT DISORDER CLASSIFICATION

Developments in motion analysis systems are distinctive in last decades. Those systems became very important tools for diagnosis of various gait disorders. They evolved so much that clinicians nowadays dare to use them in critical decisions. Thanks to advances in computer and motion capture technology, several biomechanical joint trajectories of human gait are available. Examining all parameters is wearisome and time consuming. Recent inclinations are towards facilitation of neural networks in similar cases. An Artificial Neural Network could be trained and considered as a decision support system for gait analysis.

In this study a neural network is trained for classification of four different gait patterns. Supervised learning method and Error Back-Propagation Algorithms are deployed for the training of the Multilayer Perceptron. Matlab programming language was exploited for writing the code of the algorithm. Overall 150 subjects were used in this thesis. Their age range was between six and twelve years. Samples are collected for normal gait, Right Hemiplegia, Left Hemiplegia and Diplegia from Istanbul University Istanbul Medical Faculty Motion Analysis Laboratory.

Attained classification success for distinguishing normal and for three different abnormal gaits was on average 77%. Further increase in success was achieved after the implementation of cross validation and early stopping methods, reaching at 85%.

For the classification of normal and abnormal gaits into two groups a better classification success rate was achieved, up to 96%.

There is still space to build upon the current research for further progress. This neural classifier could help clinician to support his/her decisions.

Keywords: Motion analysis, Neural network, Decision support.

ÖZET

YAPAY SİNİR AĞI ARACILIĞIYLA YÜRÜME BOZUKLUĞU SINIFLANDIRILMASI

Son zamanlar hareket analiz sistemlerinde kayda değer gelişmeler gözlenmiştir. İşbu sistemler çeşitli yürüme bozuklukların teşhislerinde önemli araç haline gelmiştir. Gelişmeler klinisyenleri kritik kararları almakta cesaretlendirmektedir. Bilgisayar ve Hareket kaydı teknolojilerindeki ilerlemeler sayesinde yürümede çeşitli biyomekanik eklem eğrisi elde edilebilmektedir. Elde edilen tüm parametrelerin incelenmesi yorucu ve zaman isteyen bir işlemdir. Son zamanlar Sinir Ağları benzer hususlarda kullanılmaya eğilimindedir. Yapay Sinir Ağı eğitilebilir ve Yürüme Analizi için Karar destek sistemi olarak düşünülebilir.

Bu çalışmada sinir ağı dört farklı yürüme şeklini sınıflandırmak için eğitilmiştir. Denetimli öğrenme metodu ve Hata Geri Yayınım algoritması Çok Katmalı Algılayıcıları eğitilmesi için kullanılmıştır. Algoritma Matlab programlama dili ile uygulama haline getirilmiştir. Bu tezde Normal Yürüme, Sağ Hemiparezi, Sol Hemiparezi ve Spastic Diparezi verileri 6 ve 12 yaşlarında toplam 150 kişiden alınmıştır. Çalışmalar İstanbul Üniversitesi İstanbul Tıp Fakültesi Hareket Analizi Laboratuarında yürütülmüştür.

Normal ve üç farklı anormal yürüme şekillerini ayırt etmede ortalama % 77 sınıflandırma başarısı elde edilmiştir. Çapraz onaylama ve eğitimi erken durdurma metoduyla daha yüksek başarım elde edilmiş ve % 85'e ulaşmıştır.

Normal ve anormal yürüme şekillerinin iki gruba sınıflandırılmasında % 96'ya varan daha yüksek başarıma ulaşılmıştır.

Mevcut araştırmada yapılan ilerlemelere rağmen daha öte gelişmeler için çalışmaların devam edilebilir. Sinir Ağı sınıflandırıcısı klinisyenlerin kararlarına destek olarak yarar sağlayabilir.

Anahtar Kelimeler: Hareket analizi, Sinirsel ağı, Karar destek.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AMP	Amplifier
ANN	Artificial Neural Network
ARS	Anatomical Reference System
BPN	Back-Propagation Network
BTS	Biotechnology Scientists
CNS	Central Nervous System
COM	Center of Mass
CP	Cerebral Palsy
EMG	Electromyography
FM	Frequency Modulation
GRF	Ground Reaction Force
GRS	Global Reference System
IR	Infra Red
LED	Light Emitting Diode
MLPBP	Multilayer Perception Back-Propagation
MSE	Mean Squared Error
NN	Neural Network
PCA	Principal Components Analysis
RF	Radio Frequency
TTE	Total Training Error
XOR	Exclusive OR

1. INTRODUCTION

Human motion analysis is a broad field with various applications and methods. It is used today for both clinical and research applications. It generally interprets biomechanics of human from the observations. Gait analysis is the part of the Human Motion analysis, which concentrates on Lower Extremity.

First observations on human motion started at ancient times. Interest over the function of movement continued for centuries. The seeds of the scientific observation rooted with the work of W. Weber and E. Weber (1836) brothers. They introduced the research of the mechanics of human walking. E. Marey (1867) had begun investigating the external motion and movement of bodies by using the photography techniques. E. Muybridge (1880s) began to use cameras to record motion during gait. W. Braune and O. Fisher (1895) added mathematical techniques to calculate the velocities, accelerations and forces during the biomechanics of gait. R. Schwartz (1933) found new methods by constructing the electrobasograph, which enable to measure the ground reactive forces directly [1].

Over the next decades, new invention technology allowed the development of the modern motion analysis laboratory. Recent advances in biomedical technology now allow accurate analysis of many specific gait characteristics such as joint angles, angular velocities and angular accelerations (kinematic analysis); ground reaction forces, joint forces, moments and powers (kinetic analysis); and electromyography (EMG).

Gait analysis is used as an important aid for decision making in different pathological cases. The aim of this project is classifying various walking conditions of collected data by the help of an artificial neural network. There are many types of pathological gaits that are due to some physical malfunction. Some are; short leg limp, antalgic limp, unstable hip limp - trendelenburg gait, stiff hip gait, stiff knee gait, gluteus maximus weakness, quadriceps weakness, muscle weakness - calcaneus gait, muscle weakness - drop foot gait, muscle spasticity or incoordination - cerebral palsy, hysterical gait, proprioceptive disturbances - ataxic gait. In this project we will be interested with cerebral palsy because, cerebral palsy is

the major field of clinical interest among several other cases and also gait analysis has played a significant role in the development of surgical treatment of children with cerebral palsy and proved to be useful in the study of walking abnormalities.

Technology supporting human motion analysis has advanced dramatically. The issue of its clinical value is related to the length of time it takes to do an interpretation which effects cost and the quality of the interpretation. Techniques from artificial intelligence such as neural networks and knowledge-based systems can help overcome these limitations. The objective of this study is to classify gait disorders by means of neural networks.

Motion analysis becomes very important tool for diagnosis of various motion disorders. Recent advances in those systems dare clinicians to use it as decision aid as well. However, examination of outputs of those systems is costly with respect of time and labor. Neural network has potential to overcome it, yet it could be used as decision support.

2. THEORETICAL BACKGROUND

2.1 Motion Analysis

Locomotion is the action with which the entire body moves through aerial (in air), aquatic or ashore. Gait Analysis system is simply a measurement system which allows us to monitor and analyze the human locomotion. The results of gait analysis have been shown to be useful in determining the best way of treatment in patients. It is also valuable after surgery to learn whether the problem has been corrected and how motion is now affecting the dynamics of walking. Through gait analysis, kinematic and kinetic data are obtained and analyzed to provide information that describes the gait characteristics. Identification of gait abnormalities visually is difficult so, a gait evaluation consists of a number of tests performed during each visit [2].

2.1.1 Motion

Since walking involves cyclical movement patterns at multiple joints, it is important to measure these kinematic patterns as a basis for interpreting other gait data (EMG, force, stride characteristics). The kinematic measurements (which also include limb segment velocities and accelerations) are necessary for the determination of joint moments and forces (kinetics).

Two basic types of motion measurement systems are in use today: electrogoniometers and video motion systems. It is more convenient to use later one because of its flexibility. Although other techniques exist such as hand digitized film, strobe light photography, and electromagnetic, they have been replaced by newer technologies.

2.1.1.1 Electrogoniometers

Electrogoniometers are electro-mechanical devices that span a joint to be measured, with attachments to the proximal and distal limb segments. In Figure 2.1 a Penny & Giles strain gauge electrogoniometer applied at the knee. The strain gauge in the small spring

measures the angle between the plastic endblocks that are attached to the leg with double-sided adhesive tape [1]. These devices provide an output voltage proportional to the angular change between the two attachment surfaces. They operate on the assumption that the attachment surfaces move with (track) the midline of the limb segment onto which they are attached, and thereby, measure the actual angular change at the joint.

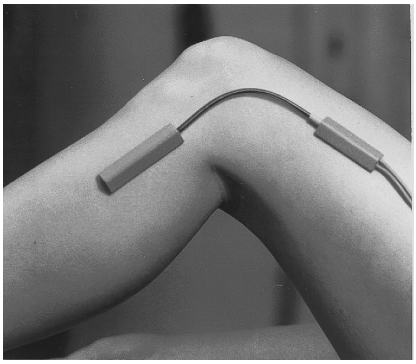


Figure 2.1 A Penny & Giles strain gauge electrogoniometer [1].

The two major advantages of these devices are low cost and ease of use. As is the case with all gait instrumentation, care must be exercised in applying them to the individual. The tracking assumption is reasonable for lean individuals, but the more "fleshy" and/or muscular the person being tested, the less likely the true angular change will be recorded due to skin and muscle movement. When considering these devices for gait, their accuracy should be carefully evaluated by testing them on individuals of various statures. The person should move through a known range of motion (i.e., 90°) while the goniometer output is being recorded. This will give a general idea of the kinds of errors the clinician might encounter.

2.1.1.2 Video Motion

Video systems utilize one or more video cameras to track bright markers placed at various locations on the person being tested. The markers are either infrared (IR) light-emitting diodes (LEDs) for active marker systems or solid shapes covered with retroreflective tape for passive marker systems. The systems keep track of the horizontal and vertical coordinates of each marker from each camera. In three-dimensional (3D) systems, the computer software computes 3D coordinates for each marker based upon the 2D data from two or more cameras and the known location of all cameras. In practice, more than two

cameras are needed, as markers become obscured from camera views because of arm swings, walking aids, and/or patient rotation.

If only one camera is used (2D), the assumption is that all motion is occurring in a plane perpendicular to the camera axis. This is seldom the case and any marker movement outside this plane will be distorted. As a result, 2D systems are not recommended for gait.

Active marker systems have LED markers that are pulsed sequentially, so the system automatically knows (by virtue of the pulse timing) the identification of each marker. Marker tracking is not a problem, since the system can maintain the identification of markers temporarily lost from view or with crossed trajectories. Merging of markers cannot occur with these systems, so the markers can be placed close together. In Figure 2.2, CODA mpx30 active, light-emitting diode (LED) motion markers placed on the foot of a subject. The sequentially strobed markers enable them to be placed close together without merging in the cameras. Each battery pack provides the power for two markers and houses circuitry to receive an infrared (IR) strobe signal. Photograph is courtesy of Charnwood Dynamics and is used with permission [1]. These systems have the disadvantage of requiring that more equipment be placed on the user. A battery pack, pulsing circuitry, and the LEDs and cables must be attached to and carried by the user. For long duration tests, heat generated by the LEDs might be a problem.

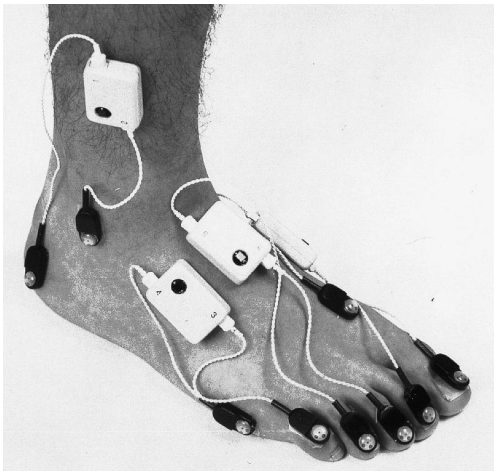


Figure 2.2 Light-emitting diode (LED) motion markers [1].

Passive marker systems have the advantage of using lightweight reflective markers without the need for electrical cables or batteries on the user. IR LEDs around each camera lens send out pulses of IR radiation that are reflected back into the lens from the markers. Passive marker system is shown in Figure 2.3. A subject walking with lightweight reflective (passive) motion markers positioned for a unilateral gait test [3]. Note the ring of IR LEDs around the camera lens. There are markers in right bottom of Figure 2.3, which have different heights [4]. IR filters are used on the camera lenses and system thresholds are set to pick up the bright markers while less bright objects in the background are suppressed. Because of their passive nature, each marker trajectory must be identified with a marker label and tracked throughout the test. When markers are lost from view or their trajectories cross, they can lose their proper identification. Sophisticated tracking software exists that does a good job; however, user intervention is sometimes required. Potential merging of markers in various camera views places limitations on how close together markers may be placed with these systems.

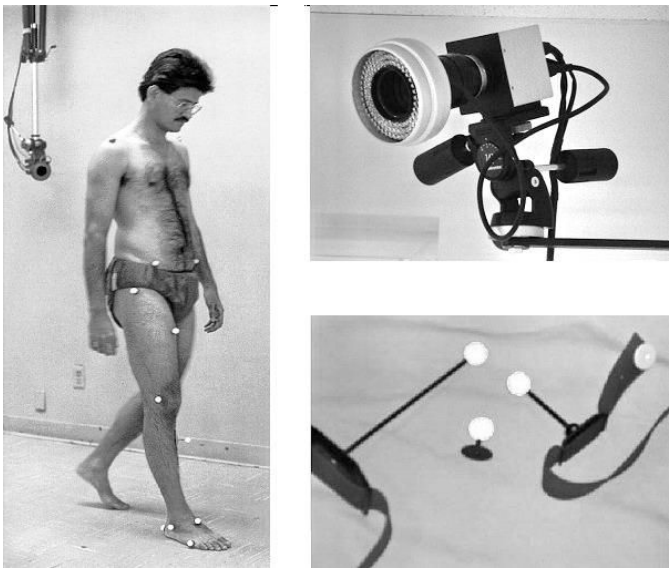


Figure 2.3 Passive marker system [3,4].

All of the systems provide the capability of acquiring at least 16 channels of analog data simultaneously with the motion data. Most compute temporal gait parameters measured from bilateral motion data if footswitches are not used. Most gait motion data are collected at

a frame rate of 50 or 60 Hz. The camera's field of view limits the number of strides available. Unlike footswitch systems, however, step length can be obtained from motion data.

2.1.2 Force

Gait is the result of muscle action exerting forces on the skeletal limb segments to produce motion and hence locomotion. It is not possible to measure these internal muscular forces. However, we can learn a lot about joint loading by measuring external forces.

2.1.1.3 Force Plates

A force plate measures the ground reaction forces exerted by a person as he or she steps on it during gait. There is illustration of exerted force in Figure 2.4. These devices consist of a top plate (mounted level with the surrounding floor) separated from a bottom frame by force transducers near each corner. Any force exerted on the top surface is transmitted through the force transducers. Force plates enable one to measure not only the vertical and shear forces, but also the "center of pressure" during gait.

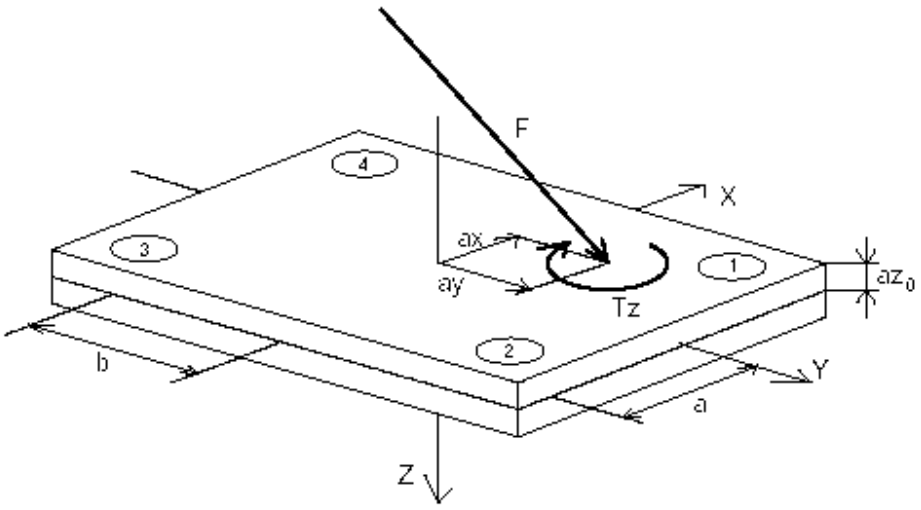


Figure 2.4 Measured force and moment components.

Two types of force plates are commercially available: piezoelectric and strain gauge. For clinical gait applications, the type probably makes very little difference. Piezoelectric force plates utilize quartz transducers, which generate an electric charge when stressed. They do not require a power supply to excite the transducers however, special charge amplifiers and low noise coaxial cables are required to convert the charge to a voltage proportional to the applied load. The transducers are calibrated at the factory and no recalibration is necessary. In general, piezoelectric force plates are more sensitive and have a greater force range than strain gauge types. They do have some slow drift, which requires resetting of the charge amplifiers just prior to data acquisition. Strain gauge force plates utilize strain gauges to measure the stress in specially machined aluminum transducer bodies (load cells) when a load is applied. They do not require the special cabling and charge amplifiers of the piezoelectric type however, they do require excitation of the strain gauge bridge circuit.

2.1.3 Electromyography

EMG is a very valuable tool in clinical gait analysis, as it can give the clinician an accurate representation of what the muscles are doing to contribute to the gait deviations observed and measured by the other instrumentation (i.e., motion, footswitches). Many surgical decisions are made based on the EMG records; therefore, it is extremely important to have instrumentation and techniques that provide high quality EMG signals. Surface electrodes have gained widespread use due to their ease of application and because skin penetration is not required. However, deep muscles can be reliably obtained only with intramuscular wire electrodes, since "cross talk" from more superficial muscles will render a surface EMG useless.

2.1.1.4 EMG Analysis Systems

Much can be learned about a person's gait by a trained clinician viewing the raw gait EMG record; however, computerized analysis systems can provide valuable assistance and make the task less tedious and time consuming. One should keep in mind, however, that computers can only work with the instructions given and the data provided. With patient data, strides can be irregular, and if the software utilizes footswitches to define the gait cycle,

problems can occur. For example, a scuff of the foot during swing may appear to the computer analysis software as another stance period. How the software handles these problems is very important. There is no substitute for a trained clinician viewing the raw record to make sure the computer analysis makes sense.

2.1.1.5 Wire EMG Electrodes

EMG Paired Hook Wire Electrodes are made of insulated nickel alloy wire. The two wires are bent approximately 180° where they exit the tip of a hypodermic needle. The bent end of one wire is 5-mm long and the other 2 mm. Both have 2 mm at the end stripped of insulation. They are available in 25 gauge, 50-mm long and 27 gauge, and 30-mm long needles. In the Figure 2.5 hypodermic needle being removed from intramuscular wire EMG electrode following insertion in the muscle. Note the loop of wire, which allows the wire to move as the muscle contracts [5].

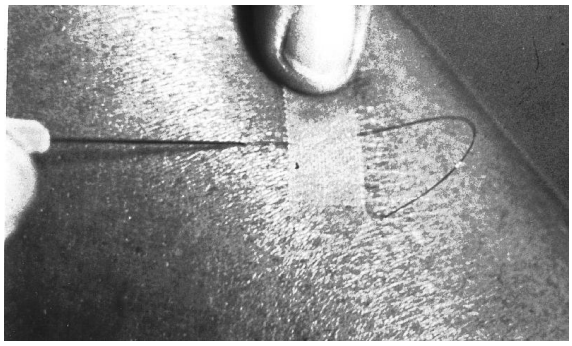


Figure 2.5 Hypodermic needle [5].

2.1.1.6 Surface EMG Electrodes

Surface electrodes come in two basic types: passive and active.

Passive electrodes are of the "Beckman silver/silver chloride" type and come as individual electrodes, so that a pair can be spaced over the muscle as desired. They are available in various sizes ranging from about 7 mm to 20 mm in diameter. Conductive electrode gel is required with these electrodes, as well as double-sided tape washers (collars), for attachment to the skin.

Active electrodes have become quite popular, as they provide signal amplification at the electrode site. A Delsys active surface EMG electrode placed over the quadriceps muscle

of a subject, show in Figure 2.6 [5]. This reduces the electrical "noise," which can be picked up by passive electrode lead wires. A number of electrodes are available, all having high impedance differential amplifier inputs with high common mode rejection ratios.

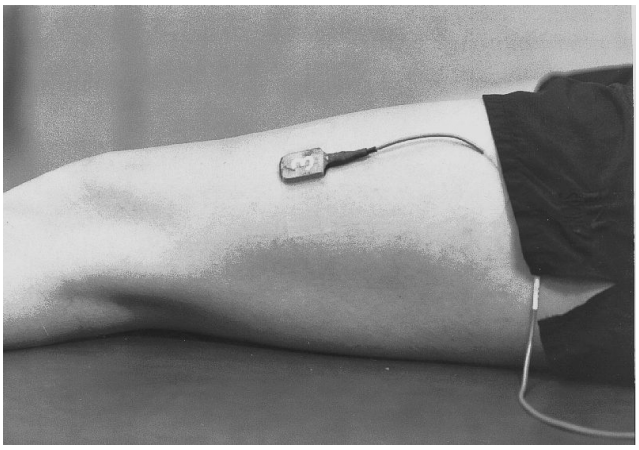


Figure 2.6 Active surface EMG electrode [5].

2.1.1.7 EMG Acquisition Systems

EMG data acquisition systems come in two types: cable and wireless. Cable systems eliminate the need for a battery on the wearer (power can be obtained through the cable) and signals are free from any radio frequency (RF) interference or dropout. In the figure below, Figure 2.7 Motion Lab's EMG system illustrated: Motion Lab Systems EMG Amplifier and

Figure 2.7 Motion Lab Systems EMG Amplifier and data acquisition system [2].

data acquisition system 3 mm dia wire cable Optional FO cable 10 EMG 2 FSws 20 to 2.3K Bandwidth Highpass (20 to 170) Lowpass (5, 10, 40, 150, 300, 600, 1.3K, 2.5K) [2]. The disadvantage is the need for a cable connecting the wearer to the instrumentation.

Wireless systems are either radio telemetry or data loggers. Telemetry systems eliminate the cable, but suffer from problems with signal dropout and RF interference. In the figure Figure 2.8 FM telemetry device of BTS is shown with 8 Channel, 2 Foot-Switchs, 5kHz Bandwidth Highpass adjustable to 1, 5, 10kHz and Lowpass adjustable to 600, 400, 200, 100 Hz. properties [4]. They also require the use of a body-worn battery. Data loggers eliminate the cable and RF problems, but require a body-worn battery and are limited in the amount of data that can be acquired before being downloaded to the computer.

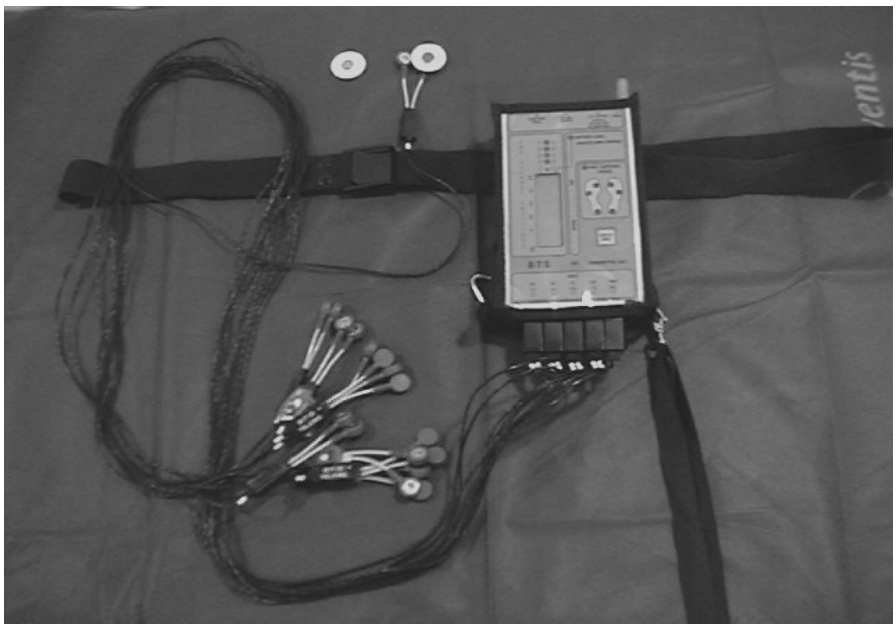


Figure 2.8 BTS FM Tele-metry (diversity receiver) [4].

2.1.4 Theory of Motion Analysis

It should be pointed out that just because a system computes the 3D coordinates of each marker, it does not mean, a priori, that 3D kinematics will be produced. To obtain true 3D motions, each body segment must be defined by at least three markers (which create a

plane passing through the segment), joint centers must be defined, and Euler angles computed. Knee and ankle joint centers are either determined from width measurements or medial markers used only during a calibration ("quiet standing") test.

Link segment model is an idealization of a non-rigid body onto a rigid frame. In this model each segment of the body is defined using 3 marked points: at two nodes apart and in the middle of the limb. These markers should form a plane on the limb and the lengths in between. Those planes are used to determine:

- The center of rotation of each joint,
- The linear and rotational position of each limb

2.1.1.8 Kinematics

Kinematic variables describe the movement, independent of the forces that cause the movement. They include linear displacements, *d*, speed, *s*, and velocities, *v*, and accelerations, *a*, and also angular displacement, Θ, velocities, ω, and accelerations, α [7].

Movement or motion indicates a joint action, the relative angular movement of the limbs on the distal and proximal sides of the joint [7]. By identifying the plane of motion in which direction the action takes place, motions are defined by flexion and extension occurring when movement around the transversal axes, movement around anteroposterior axes is identified by abduction (moving out from the body) and adduction (toward the body), movement around the longitudinal axes is called rotation, internal and external.

To describe the kinematics of a segment a coordinate reference system is used. Each segment can be set with an origin and a principal axis, which is usually defined along the long axis of the segment. Three dimensional imaging systems are used to detect kinematic data from a subject in a motion analysis. Three types of coordinate systems are used to derive the kinematics, the global reference system (GRS) which remains fixed in space, a technical reference system which is derived from the subject's reflective markers, and an anatomical reference system (ARS) which is attached to each body segment [6].

A moving segment's translation and rotation, are described by the segment's ARS position and orientation relative to the GRS. In two dimensional translation of a point, a, in the ARS, (x_i,y_i), relative to the GRS, (X_i, Y_i), can be written as [7]:

$$\begin{bmatrix} X \\ Y \end{bmatrix}_a = \begin{bmatrix} X \\ Y \end{bmatrix}_i + \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_{ia} \qquad (2.1)$$

Where *s* is denoting sine and *c* cosine of an angle (θ).

A rigid body moving in a three dimensional system, has a possibility of translations and rotations. This implies a 3 × 3 transformation matrix, [ϕ]. A common angular system used to define the angular orientation in space is the Euler system of angles. A specific definition of Euler angles will describe in which sequence the rotations take place [7]. Euler's convention contains 12 different sequences of rotations [8].

An axis system denoted by *x, y, z* will be transformed into a system denoted by *x‴,y‴,z‴* in a chosen sequence. The Cardan system, *x*−*y*−*z*, which is common in biomechanics, describes the sequence order of axes to rotate about (Equations 2.2 and 2.3) [7]. The first rotation, θ_1, is about the x axis to get *x′,y′,z′;* the second rotation, θ_2, is about the new y' to get *x″, y″, z″;* and the last rotation, θ_3, is about *z″* to get to *x‴,y‴,z‴*.

An assumed point in the original *x, y, z* axis system with coordinates x_0, y_0, z_0, will have coordinates x_1,x_1, x_1 in the *x′,y′,z′* axis system, based on rotation θ_1. The second rotation θ_2 will provide coordinates x_2,y_2,z_2 in the axis system *x″,y″,z″,* and the final rotation θ_3 will provide the point the coordinates x_3, y_3, z_3 in the axis system *x‴,y‴,z‴*. These rotation can be written:

$$[\phi_1] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_1 & s_1 \\ 0 & -s_1 & c_1 \end{bmatrix}, \quad [\phi_2] = \begin{bmatrix} c_2 & 0 & -s_2 \\ 0 & 1 & 0 \\ s_2 & 0 & c_2 \end{bmatrix}, \quad [\phi_3] = \begin{bmatrix} c_3 & s_3 & 0 \\ -s_3 & c_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (2.2)$$

where (c_1, s_1), (c_2, s_2) and (c_3, s_3) denote ($\cos\theta_1$, $\sin\theta_1$), ($\cos\theta_2$, $\sin\theta_2$), and ($\cos\theta_3$, $\sin\theta_3$) respectively.

The transformation of the point from the axis system *x, y, z* to *x‴,y‴,z‴* can then be written [8]:

$$\begin{bmatrix} x''' \\ y''' \\ z''' \end{bmatrix} = [\phi_3] \cdot [\phi_2] \cdot [\phi_1] \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.3}$$

2.1.1.9 Kinetics

Kinetics deals with the forces associated with a movement. Internal forces come from muscle activity, tendons and ligaments, and joint contact forces and external forces comes from ground reaction forces, segment weight, or applied loading on the musculoskeletal system [7]. Knowledge of the muscle forces is important for understanding the causes of movement. Kinetic quantities are evaluated from kinematics, anthropometric data and external forces, such as the ground reaction force, f_{GRF}, [7].

Newton's second law of motion (Equation 2.4) together with the Euler dynamic equation (Equation 2.5), make up the basis for the mathematical model of the limbs called link-segment modelling using inverse dynamics, where reaction forces, *R,* and muscle moments, *M,* are calculated [7,9]:

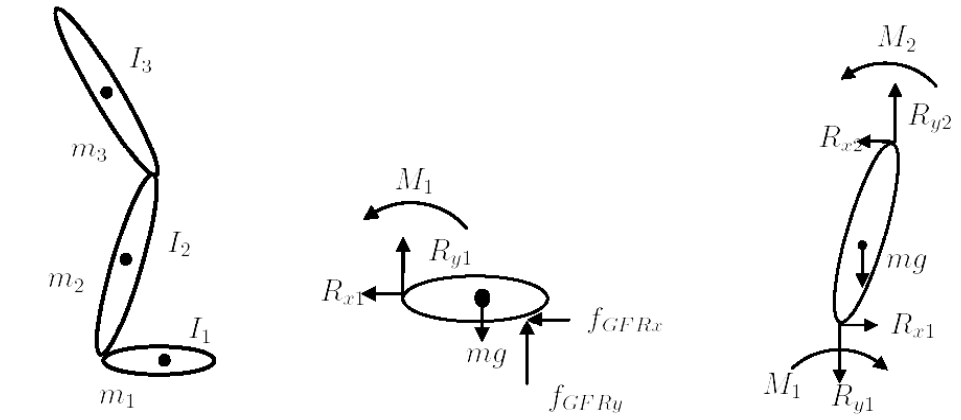
$$R = m \cdot a \tag{2.4}$$

$$M = I \cdot \alpha \tag{2.5}$$

where m is the mass of the object, a is the linear acceleration, I, is the mass moment of inertia and α is the angular acceleration of the object.

Link-segment modelling and free body diagram of the foot- and shank segment, showing moments of inertia, I_i, masses, m_i, reaction forces, R_{yi} and R_{xi}, joint moments, M_i, i = 1,2,3. Moment and forces are calculated via inverse dynamics (Equations 2.6 and 2.7). The link-segment model is broken down at the joints into segments which are treated separately as rigid bodies, creating a free-body diagram (Figure 2.9(a)) [10]. In accordance to Newton's third law, there is an equal and opposite force acting at each joint, and moment and forces can be evaluated at any joint with a known external loading or reaction force. Modeling the limb in this way carries the assumptions that (1) each segment has a fixed mass located at its center of mass (in the center of gravity) (2) the location of each segment's center of mass (COM)

remains fixed (3) joints are considered without translations (4) mass moment of inertia of each segment is constant and (5) the length of each segment is constant during movement [7].



(a) Link Segment (b) Foot Segment (c) Shank Segment

Figure 2.9 Link-segment modelling and free body diagram.

The method of inverse dynamics is usually employed in gait analysis to compute the net joint moments, net joint powers and intersegmental forces. Evaluation starts at the foot segment (Figure 2.9(b)) with the ankle joint forces and moment (Equations 2.6 and 2.7) [10]. Establishing the forces in the y direction and the moment about the ankle, M_i:

$$\sum R_{yi} = m_i \cdot a_{yi} = f_{GRFyi} + R_{yi} - mg \qquad (2.6)$$

$$\sum M_i = I_i \cdot \alpha = f_{GRFyi} d_{GRF} + R_{yi} d_{yi} \qquad (2.7)$$

where d_{GRF} describe the moment arm of the ground reaction force f_{GRFy} to the ankle joint and d_y the moment arm to the reaction force R_{y1} to the ankle joint. The carry weight mg is located at the center of mass (COM), moment of inertia, I, is evaluated from anthropometric tables. Linear and angular acceleration, a and α, are obtained from kinematic data. From this point the knee joint moment and forces can be evaluated by applying equal and opposite reaction force on the shank segment (Figure 2.9(c)) [10].

2.2 Gait Abnormalities

Walking or gait abnormalities are unusual and uncontrollable problems with walking. Many different types of gait abnormalities are produced unconsciously. Most, of them are due to some physical defect [14].

2.2.1 Normal Walking

Normal Walking Requires: good locomotor generator, two legs with the same length and joints should not be stiff.

2.2.2 Cerebral Palsy

Cerebral palsy (CP) is a group of disorders characterized by loss of movement or loss of other nerve functions. "Cerebral" refers to the brain, and "palsy" is a disorder of movement or posture. These disorders are caused by injuries to the cerebrum (the largest portion of the brain) that occur before, at, or within 5 years of birth [15].

Cerebral palsy is the term for a range of non-progressive syndromes of posture and motor impairment that results of an insult to the developing central nervous system. It is the most common cause of severe physical disability in childhood. The worldwide prevalence and incidence of the disorder are not clearly known. The overall reported prevalence in children aged 3-10 years is 2.4 per 1000 children, with variability in the reported rates in girls and boys. During the past 20 years, there have been increases in the incidence and prevalence of cerebral palsy [22].

Cerebral palsy is characterized by an inability to fully control muscles, movement, and posture. Depending on the extent of brain involvement, it can range from mild to severe. A person with cerebral palsy may also have problems with vision, speech, nutrition, feeding, and hearing; and may experience seizures, developmental delay, and learning disabilities. However, many people with cerebral palsy lead healthy productive lives.

Today doctors classify cerebral palsy in to four broad categories as;

- Spastic,
- Dyskinetic (Athetoid),
- Ataxic,
- Mixed Cerebral Palsy.

Spastic CP is the most common form and affects the body's ability to relax muscles, causing tightness and difficulties in movement. It may affect a single limb; one side of the body (spastic hemiplegia), both legs (spastic diplegia) and both arms and legs (spastic quadriplegia). Athetoid cerebral palsy affects the ability to control muscles, leading to involuntary and uncontrolled movements in the affected muscles. Children with Ataxic CP have a disturbed sense of balance and depth perception, characterized by tremors or shaky movements. Each case of cerebral palsy affects a child differently, and some have more than one form of CP (Mixed cerebral palsy).

Spastic cerebral palsy includes about 50% of cases. Dyskinetic (athetoid) cerebral palsy affects about 20%. It involves development of abnormal movements (twisting, jerking, or other movements). Ataxic cerebral palsy involves tremors, unsteady gait, loss of coordination, and abnormal movements. It affects about 10%. The remaining 20% are classified as mixed, with any combination of the above [15].

Cerebral classified according to the part of the body affected in Figure 2.10:

- Hemiplegia: affecting one side of the body;
- Diplegia: affecting the legs more than the arms;
- Triplegia: affecting three extremities;
- Asymmetric diplegia: affecting one side of the body more than the other side;
- Quadriplegia: affecting all four extremities;
- Total: affecting all extremities and multiple other body systems;

In our thesis we will try to identify hemiplegia, loss of muscle control on one side of the body which has been caused by damage to the opposite side of the brain, and diplegia, all four limbs affected but lower limbs more affected than upper limbs.

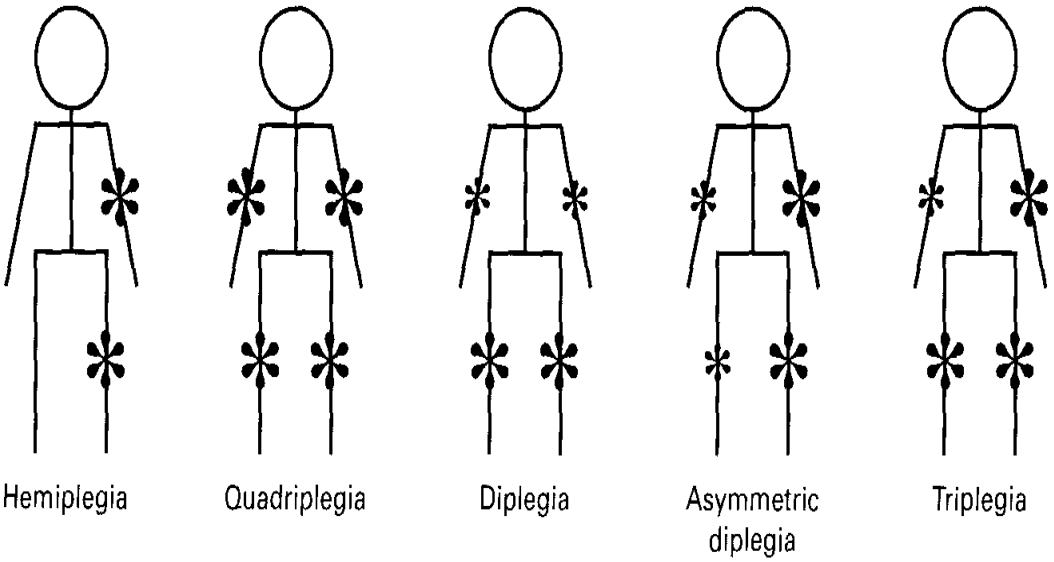


Figure 2.10 Classification of Cerebral Palsy according to the effected part of the body.

2.3 Neural Networks

2.3.1 Basics Of Neuronal Morphology

An average brain contains ~100 billion neurons, each of which has 1000–10 000 connections with other neurons. Neurons consist of a cell body which includes nucleus that controls the cell activity, many fine treads, dendrites, that carry information into the cell, and one longer thread known as the axon which carries the signal away, Figure 2.11.

Impulses pass along the axon to the synapse, the junction between one neuron and the next and signals are passed from one to the next in an all-or-none fashion. Neurons are organized in a fully connected network and act like messenger in receiving and sending impulses. The result is an intelligent brain capable of learning, prediction and recognition.

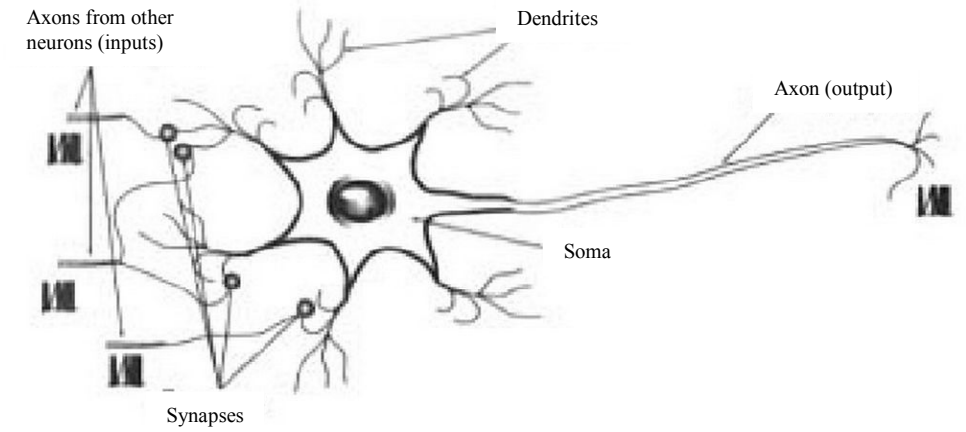


Figure 2.11 Biological neuron: a biological computing processing element.

Nature has developed a very complex neuronal morphology in biological species. Biological neurons, over one hundred billion in number, in the central nervous systems (CNS) of humans play a very important role in the various complex sensory, control, affective, and cognitive aspects of information processing and decision making. In neuronal information processing, there are a variety of complex mathematical operations and mapping functions that act in synergism in a parallel cascade structure forming a complex pattern of neuronal layers evolving into a sort of pyramidal pattern. The information flows from one neuronal layer to another in the forward direction with continuous feedback, and it evolves into a dynamic pyramidal structure. The structure is pyramidal in the sense of the extraction and convergence of information at each point in the forward direction. A study of biological neuronal morphology provides not only a clue but also a challenge in the design of a realistic cognitive computing machine — an intelligent processor.

From the neurobiological as well as the neuralmathematical point of view, we identify two key neuronal elements in a biological neuron: the synapse and the soma. These two elements are responsible for providing neuronal attributes such as learning adaptation knowledge (storage or memory of past experience), aggregation, and nonlinear mapping operations on neuronal information. In Figure 2.12 biological neural network is illustrated.

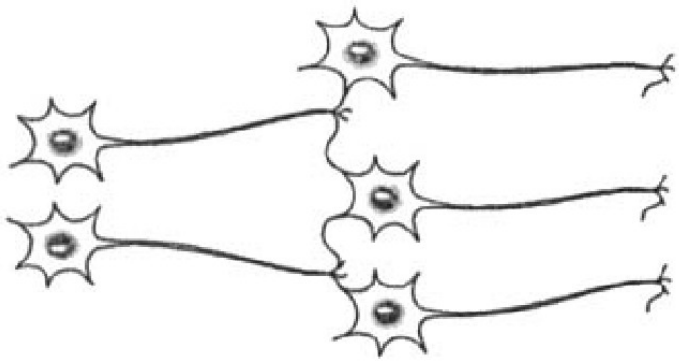


Figure 2.12 Biological neural network.

A neuron is an information-processing unit that is fundamental to the operation of a neural network. The block diagram in Figure 2.13 shows the model of a neuron, which forms the basis for designing (artificial) neural networks [11].

Here we identify three basic elements of the neuronal model:

1. A set of synapses or connecting links, each of which is characterized by a weight or strength of its own. Specifically, a signal x_j at the input of synapse j connected to neuron k is multiplied by the synaptic weight w_{kj}. It is important to make a note of the manner in which the subscripts of the synaptic weight w_{kj} are written. The first subscript refers to the neuron in question and the second subscript refers to the input end of the synapse to which the weight refers. Unlike a synapse in the brain, the synaptic weight of an artificial neuron may lie in a range that includes negative as well as positive values.
2. An adder for summing the input signals, weighted by the respective synapses of the neuron; the operations described here constitute a linear combiner.
3. An activation function for limiting the amplitude of the output of a neuron. The activation function is also referred to as a squashing function in that it squashes (limits) the permissible amplitude range of the output signal to some finite value. Typically, the normalized amplitude range of the output of a neuron is written as the closed unit interval [0,1] or alternatively [-1,1].

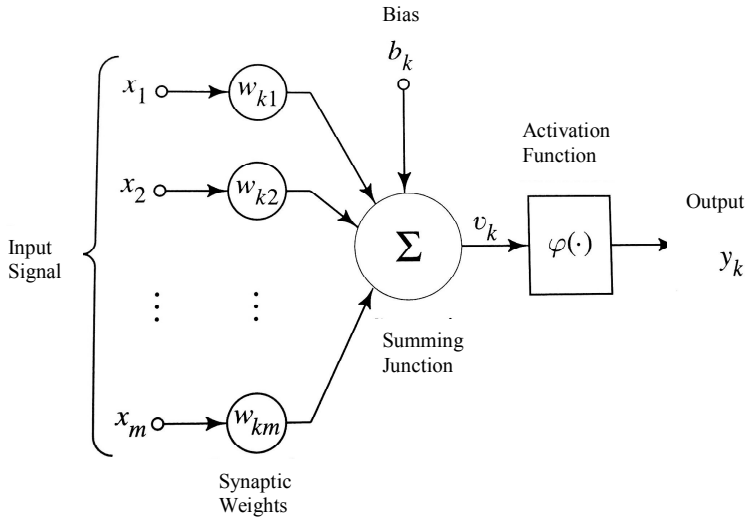


Figure 2.13 Artificial neuron: an artificial computing processing element.

The neuronal model of Figure 2.13 also includes an externally applied bias, denoted by b_k. The bias b_k has the effect of increasing or lowering the net input of the activation function, depending on whether it is positive or negative, respectively [12].

In mathematical terms, we may describe a neuron k by writing the following pair of equations:

$$u_k = \sum_{j=2}^{m} w_{kj} x_j \qquad (3.1)$$

and

$$y_k = \varphi(u_k + b_k) \qquad (3.2)$$

where x_1, x_2, ... , x_m are the input signals; w_{k1}, w_{k2}, ... , w_{km} are the synaptic weights of neuron k; u_k is the linear combiner output due to the input signals; b_k is the bias; $\varphi(\bullet)$ is the activation function; and y_k is the output signal of the neuron. The use of bias b_k has the effect of applying an affine transformation to the output u_k of the linear combiner in the model of Figure 2.13,

as shown by

$$v_k = u_k + b_k \tag{3.3}$$

In particular, depending on whether the bias b_k is positive or negative, the relationship between the induced local field or activation potential v_k of neuron k and the linear combiner output u_k is modified

2.3.2 Network Architectures

The manner in which the neurons of a neural network are structured is intimately linked with the learning algorithm used to train the network. We may therefore speak of learning algorithms (rules) used in the design of neural networks as being structured. The classification of learning algorithms is considered in the next chapter, and the development of different learning algorithms is taken up in subsequent chapters of the book. In this section we focus our attention on network architectures (structures).

In general, we may identify three fundamentally different classes of network architectures.

2.1.1.10 Single-Layer Feedforward Networks

In a layered neural network the neurons are organized in the form of layers. In the simplest form of a layered network, we have an input layer of source nodes that projects onto an output layer of neurons (computation nodes), but not vice versa. In other words, this network is strictly a. feedforward or acyclic type [12]. It is illustrated in Figure 2.14 for the case of four nodes in both the input and output layers. Such a network is called a single-layer network, with the designation "single-layer" referring to the output layer of computation nodes (neurons). We do not count the input layer of source nodes because no computation is performed there.

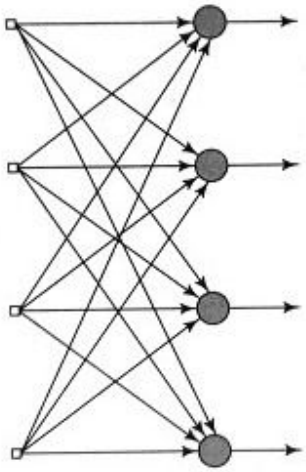


Figure 2.14 Feedforward or acyclic network with a single layer of neurons [12].

2.1.1.11 Multilayer Feedforward Networks

The second class of a feedforward neural network distinguishes itself by the presence of one or more hidden layers, whose computation nodes are correspondingly called hidden neurons or hidden units. The function of hidden neurons is to intervene between the external input and the network output in some useful manner. By adding one or more hidden layers, the network is enabled to extract higher-order statistics. In a rather loose sense the network acquires a global perspective despite its local connectivity due to the extra set of synaptic connections and the extra dimension of neural interactions [13]. The ability of hidden neurons to extract higher-order statistics is particularly valuable when the size of the input layer is large [12].

The source nodes in the input layer of the network supply respective elements of the activation pattern (input vector), which constitute the input signals applied to the neurons (computation nodes) in the second layer (i.e., the first hidden layer). The output signals of the second layer are used as inputs to the third layer, and so on for the rest of the network. Typically the neurons in each layer of the network have as their inputs the output signals of the preceding layer only. The set of output signals of the neurons in the output (final) layer of the network constitutes the overall response of the network to the activation pattern supplied

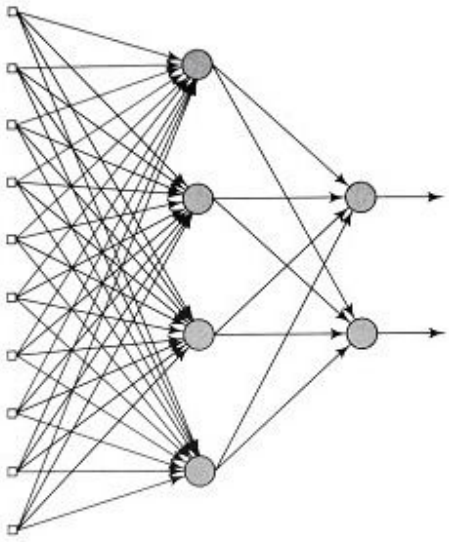


Figure 2.15 Feedforward network with one hidden and one output layer [12].

by the source nodes in the input (first) layer. The architectural graph in Figure 2.15 illustrates the layout of a multilayer feedforward neural network for the case of a single hidden layer. For brevity the network in Figure 2.15 is referred to as a 10-4-2 network because it has 10 source nodes, 4 hidden neurons, and 2 output neurons [12]. Notation 10-4-2 refers to network design.

The neural network in Figure 2.15 is fully connected in the sense that every node in each layer of the network is connected to every other node in the adjacent forward layer. If, however, some of the communication links (synaptic connections) are missing from the network, we say that the network is partially connected.

2.1.1.12 Recurrent Networks

A recurrent neural network distinguishes itself from a feedforward neural network in that it has at least one feedback loop. For example, a recurrent network may consist of a single layer of neurons with each neuron feeding its output signal back to the inputs of all the other neurons, as illustrated in the architectural graph in Figure 2.16 In the structure depicted in this figure there are no self-feedback loops in the network; self-feedback refers to a situation where the output of a neuron is fed back into its own input. The recurrent network illustrated in Figure 2.16 also has no hidden neurons. Recurrent networks with hidden neurons are also possible.

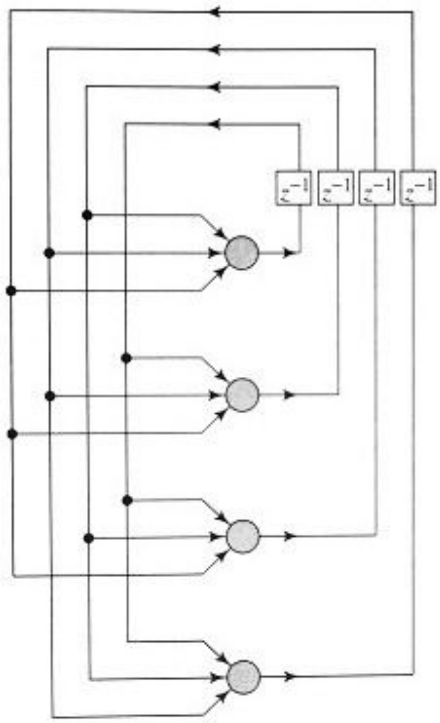


Figure 2.16 Recurrent network with no self-feedback loops and no hidden neurons [12].

Moreover, the feedback loops involve the use of particular branches composed of unit-delay elements (denoted by z^{-1}), which result in a nonlinear dynamical behavior, assuming that the neural network contains nonlinear units.

2.3.3 NN Models and Learning Algorithm

There are many different types of NNs, some of which are more popular than others. When neural networks are used for data analysis, it is important to distinguish between ANN models (the network's arrangement) and NN algorithms (computations that eventually produce the network outputs). Once a network has been structured for a particular application, that network is ready to be trained [18]. There are two approaches to training, supervised and unsupervised.

2.1.1.13 Networks with Unsupervised Learning

In unsupervised training, the network is provided with inputs but not with desired outputs. The system itself must then decide what features it will use to group the input data. This is often referred to as self-organization or adaptation. The self-organizing behavior may involve competition between neurons, co-operation or both. Neurons are organized into groups of layers. In competitive learning, neurons are grouped in such a way so that when one neuron responds more strongly to a particular input it suppresses or inhibits the output of the other neurons in the group. In co-operative learning the neurons within each group work together to reinforce their output. The training task is to group together patterns that are similar in some way, extract features of the independent variables and come up with its own classifications for inputs. NNs consider the data they are given, discover some of the properties of the data set and learn to reflect these properties in their output. The goal is to construct feature variables from which the observed variables, both input and output variables, can be predicted. Feature- extracting networks can be regarded as principal component analyses (PCA). They are used as an alternative to classical PCA for data reduction purposes, to transform the data set into a new space with retained information in data set but with a reduced number of variables (dimensionality). The goal is to construct a network that will map the entire training data (both inputs and outputs variables) at once.

2.1.1.14 Supervised Learning

The goal in supervised learning is to predict one or more target values from one or more input variables. Supervised learning is a form of regression that relies on example pairs of data: inputs and outputs of the training set.

This type of network is a system of fully interconnected neurons organized in layers, the input layer, the output layer, and the hidden layers between them. The input layer neurons receive data from a data file. The output neurons provide ANN's response to the input data. Hidden neurons communicate only with other neurons. They are part of the large internal pattern that determines a solution to the problem. Theory says that most functions can be approximated using a single hidden layer.

The information that is passed from one processing element to another is contained within a set of weights. Some of the interconnections are strengthened and some are

weakened, so that a neural network will output a more correct answer. The most commonly used learning algorithm is back propagation of error. The error in prediction is fed backwards through the network to adjust the weights and minimize the error, thus preventing the same error from happening again. This process is continued with multiple training sets until the error is minimized across many sets. This results in the mapping of inputs to outputs via an abstract hidden layer.

The number of neurons in the hidden layer influences the number of connections. During training phase inputs are adjusted (transformed) by the connection weights. Therefore, the number of connections has a significant effect on the network performance. Too few hidden neurons will hinder the learning process and too many will depress prediction abilities through over training. By increasing the number of the hidden neurons the NN more closely follows the topology of the training data set. However, exceeding an optimum number results in tracing the training pattern too closely.

When the NN produces the desired output, (i.e. is trained to a satisfactory level) the weighted links between the units are saved. These weights are then used as an analytical tool to predict results for a new set of input data. This is a recall or prediction phase when network works only by forward propagation of data and there is no backward propagation of error. The output of a forward propagation is the predicted model for the validation data.

Pattern association is usually supervised learning. NNs compete well with statistical methods in pattern recognition, especially when the systems contain high level of noise and variation.

2.3.4 Learning Rule

There are many different learning rules but the most often used is the Delta rule or Backpropagation rule. A neural network is trained to map a set of input data by iterative adjustment of the weights. The use of the weighted links is essential to the ANN's recognizing abilities. Information from inputs is fed forward through the network to optimize the weights between neurons. Optimization of the weights is made by backward propagation of the error

during training or learning phase. The ANN reads the input and output values in the training data set and changes the value of the weighted links to reduce the difference between the predicted and target values. The error in prediction is minimized across many training cycles until network reaches specified level of accuracy.

As mentioned before, there are two approaches to training, supervised and unsupervised. The most often used NN is a fully connected, supervised network with back propagation learning rule. In this study, Multilayer Back-Propogated NN is used.

2.3.5 Multilayer Back-Propogation NN and Algorithm

Multilayer perceptrons have been applied successfully to solve some difficult and diverse problems by training them in a supervised manner with a highly popular algorithm known as the error back-propagation algorithm. This algorithm is based on the error-correction learning rule.

Basically, error back-propagation learning consists of two passes through the different layers of the network: a forward pass and a backward pass. In the forward pass, an activity pattern (input vector) is applied to the sensory nodes of the network, and its effect propagates through the network layer by layer. Finally, a set of outputs is produced as the actual response of the network. During the forward pass the synaptic weights of the networks are all fixed. During the backward pass, on the other hand, the synaptic weights are all adjusted in accordance with an error-correction rule.

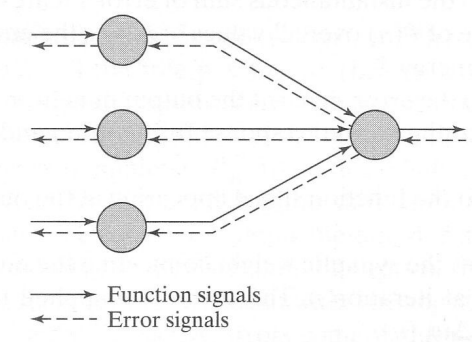


Figure 2.17 Signal flows in a multilayer perceptron[12].

Specifically, the actual response of the network is subtracted from a desired (target) response to produce an error signal. This error signal is then propagated backward through the network, against the direction of synaptic connections—hence the name "error back-propagation" [16]. Figure 2.17 above is illustration of the directions of two basic signal flows in a multilayer perceptron: forward propagation of function signals and back-propagation of error signals.

The synaptic weights are adjusted to make the actual response of the network move closer to the desired response in a statistical sense. The learning process performed with the algorithm is called back-propagation learning.

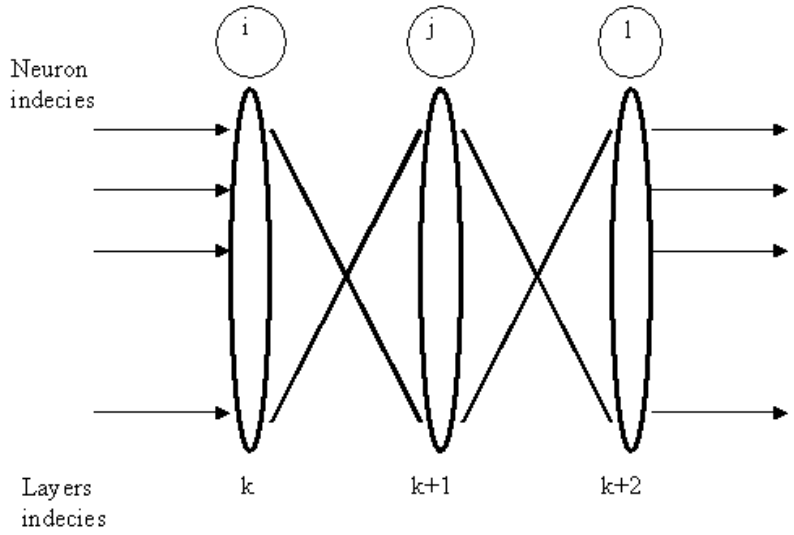


Figure 2.18 Block Diagrom of MLPBP NN.

For an output layer:

$$\Delta\omega_{ij}^{k} = \eta\delta_{j}^{k+1}o_{i}^{k}\ ,\quad \delta_{j}^{k} = -\frac{\partial E^{k}}{\partial s_{j}^{k}} \tag{4.1}$$

For any hidden layer:

$$\Delta\omega_{ij}^{k} = \eta\delta_{j}^{k+1}o_{i}^{k}\ ,\quad \delta_{j}^{k+1} = \left(\sum_{l}\delta_{l}^{k+2}\omega_{jl}^{k+1}\right)f'\left(s_{j}^{k+1}\right) \tag{4.2}$$

We define layer index as ‘k’. The net input to the i th unit on the k th layer is s_i^k. Output unit at the k th layer is o_i^k, where $o_i^k = f(s_i^k)$. Function $f(\bullet)$ is smooth and nonlinear (i.e sigmoid). Weight associated with the connection between i th unit of k th layer and the j th unit of (k+1) th layer is defined as w_{ij}^k. We ssume model with n inputs and m outputs hence we have a desired signal vector at the output defined as transpose of t = [t1, t2... tm]. MLP generates an input-output map for weights. Notation η is learning rate and δ is local gradient.

2.1.1.15 <u>Activation Function</u>

The computation of δ for each neuron of the multilayer perceptron requires knowledge of the derivative of the activation function associated with that neuron. For this derivative to exist, we require the function $f(\bullet)$ to be continuous. In basic terms, differentiability is the only requirement that an activation function has to satisfy.

An example of a continuously differentiable nonlinear activation function commonly used in multilayer perceptrons is sigmoidal nonlinearity; two forms are described:

1. Logistic Function. This form of sigmoidal nonlinearity in its general form is defined by

$$\varphi(v(n)) = \frac{1}{1+\exp(-av(n))} \quad a > 0 \text{ and } -\infty < v(n) < \infty \qquad (4.3)$$

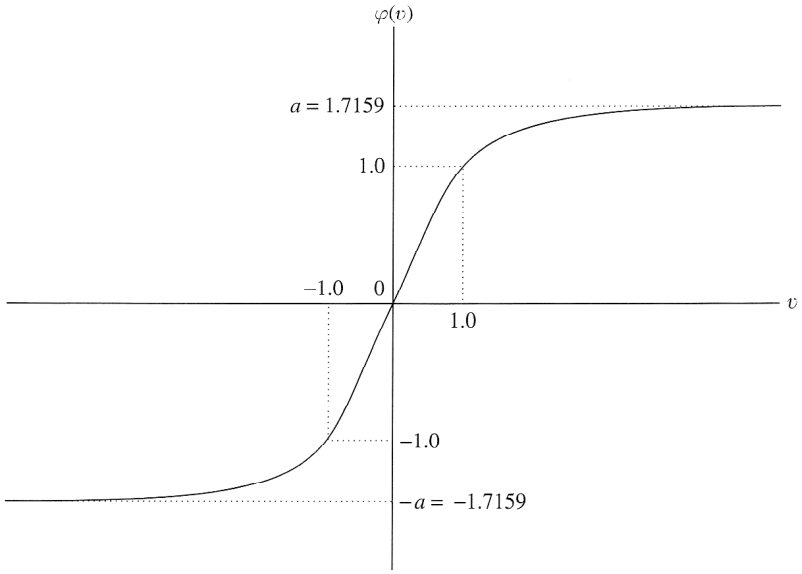


Figure 2.19 Logistic Function [12].

2. Hyperbolic tangent function. Another commonly used form of sigmoidal non-linearity is the hyperbolic tangent function, which in its most general form is defined by

$$\varphi(v(n)) = a\tanh(bv(n)), \quad (a,b) > 0 \qquad (4.4)$$

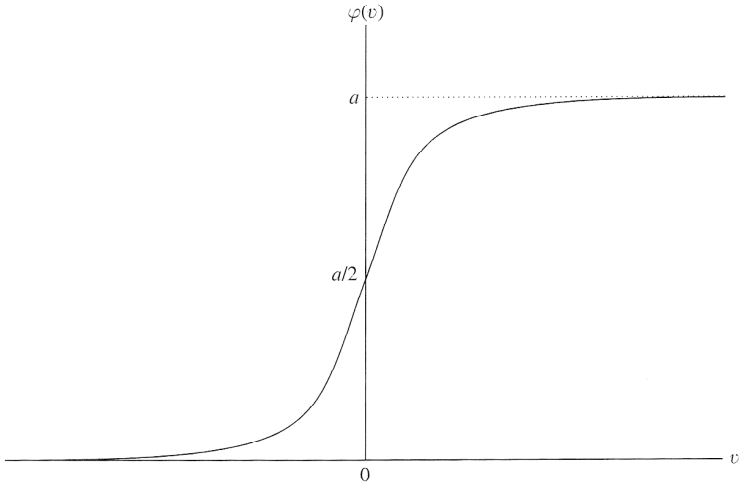


Figure 2.20 Hyperbolic Tangent Function [12].

2.1.1.16 Rate of Learning

The back-propagation algorithm provides an "approximation" to the trajectory in weight space computed by the method of steepest descent. The smaller we make the learning-rate parameter η, the smaller the changes to the synaptic weights in the network will be from one iteration to the next, and the smoother will be the trajectory in weight space. This improvement, however, is attained at the cost of a slower rate of learning. If, on the other hand, we make the learning-rate parameter η too large in order to speed up the rate of learning, the resulting large changes in the synaptic weights assume such a form that the network may become unstable (i.e., oscillatory). [12].

2.1.1.17 Momentum Term

A simple method of increasing the rate of learning yet avoiding the danger of instability is to modify the delta rule by including a momentum term:

$$\Delta\omega_{ij} = \alpha\Delta\omega_{ij} + \eta\delta_j o_i \qquad (4.5)$$

The inclusion of momentum in the back-propagation algorithm has a stabilizing effect in directions that oscillate in sign. The incorporation of momentum in the back-propagation algorithm represents a minor modification to the weight update, yet it may have some beneficial effects on the learning behavior of the algorithm. The momentum term may also have the benefit of preventing the learning process from terminating in a shallow local minimum on the error surface.

3. MATERIALS AND METHODS

The objective of this study is to design and train Neural Network that classifies four different gait patterns. Architectural design of it is accomplished according to the problem inclination. Training of the Neural Network required huge amount of samples of gait patterns. For this purpose, motion analysis system [4] with six TV cameras sensitive at infrared range was used for the data acquisition. Studies are performed at Istanbul University Istanbul Medical Faculty, Istanbul, Turkey. We have recorded subjects who are able to walk and have proper anthropometrics that motion systems could evaluate.

3.1 Procedures and Data Acquisition

First of all, some information from subject is taken.

- Subject's Personal information

Personal information is necessary to create personal database of the individual. Those are name, surname, middle name, birthday, phone number, gender and patient code. Date of acquisition performed assigned automatically. Subject's data can be searched or called according to those information.

- Physical Evaluation

The subjects were evaluated to take baseline measurements, which are active and passive joint ranges of movement before and after the treatment. Joint angle was measured by using a goniometer and the muscle force was evaluated using the gross motor test by a therapist. Those evaluations compared with report of Locomotion Analysis System for measurement consistency.

- Subject' Anatomic measurements

Subject is asked to take of his/her clothes. Some anthropometric measurements are taken, Those are weight, height, pelvis width, pelvis height, knee diameter, ankle diameter and total leg length. Antropometric measurements are indispensable for post processing procedure. Namely, they are used to create joint segment model and calculation of both kinetics and kinematics.

- Marker set up.

Retroreflective markers were placed onto the segments and joints according to Davis protocol, see Figure 3.1. Markers were attached to the bilateral lower extremities with stickers and elastic tapes on the following anatomic locations: cervical 7, sacrum, trochanter major, fibula head, anterior superior iliac, lateral malleolus, base of the heel, lateral aspect of midthigh, midcalf, fifth foot metatars, lateral femoral condyle, acromioclavular joint The detailed marker positions are given in Appendix A.

Figure 3.1 Davis model based marker setup in Motion Analysis Laboratory [4].

- Standing Record for rigid body Davis model

22 retro reflective markers are placed on the specified point of the subject. A static calibration data was collected before the walking trials with bilateral heel markers set to define the angular and special offset. Subject faced to the positive direction of the walking mile on the Force plate 1 which is the nearest to the reference frame of the Lab.

- Tracking of the standing trial

Tracklab program is used for labeling of the markers immediately after steady trial acquisition. Although it is sophisticated tracking software and does a good job user intervention is sometimes required. In the Figure 3.2 below checking markers continuity within specified period of time is performed.

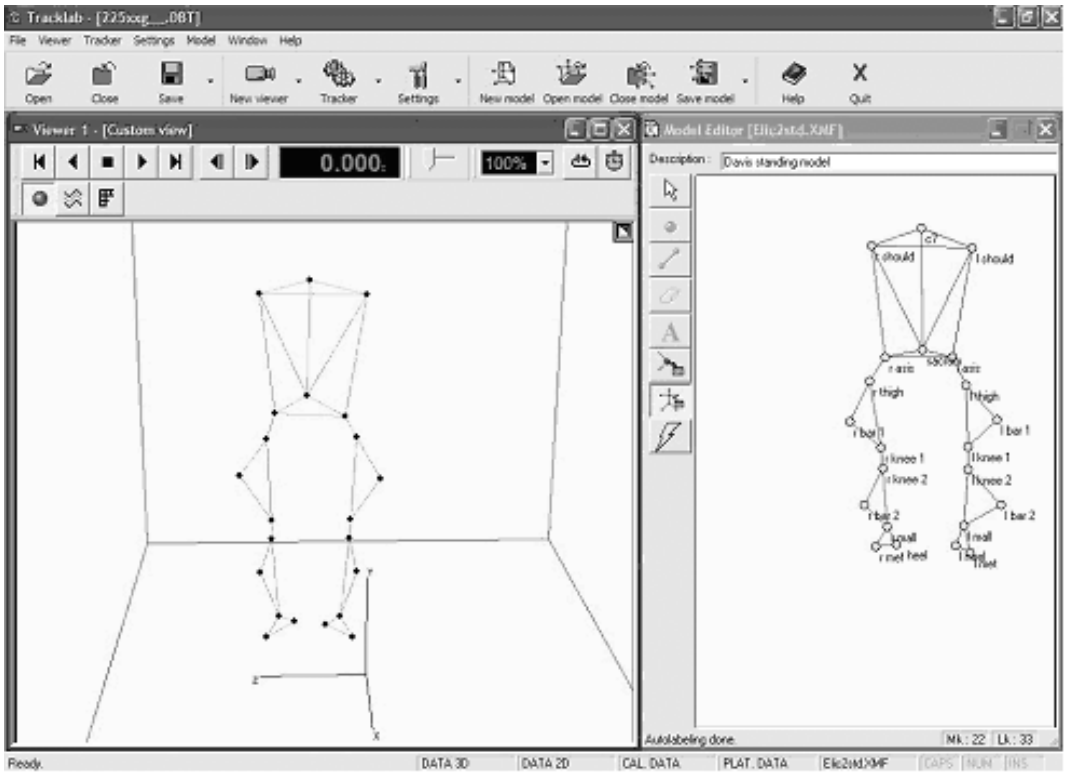


Figure 3.2 Tracking of the acquired standing trial.

- Processing of tracked data for standing trial

Since steady (standing) trial is acquired once it is very important that tracked data has no problem. Therefore data processing is accomplished shortly after tracking. In that step data filtering and interpolation are performed if they are necessary. Some critical parameters visualized by means of kinematic visualization property of program. Visualized parameters are bilateral angle of ankles and knees. Those are compared with angles that measured previously at the physical evaluation of the subject.

- Record of walking trials

After processing data it is beneficial to check bar markers alignment since most of the case hands of the subject may strike to those rigid bars. Before walking trial acquisition the heel alignment markers were removed. Direction of the movement must be appointed. Figure 3.3 is the snapshot of walking trial Acquisition in Istanbul University Istanbul Medical Faculty Motion Analysis Laboratory. Several trials are taken from each subject. During trial acquisition video are recorded to digital video camera. Those real video capture are compared with report of Motion Analysis System.

Figure 3.3 Recording of walking trial in Motion Analysis Laboratory.

- Tracking and processing of walking data

All markers should be placed at their right positions throughout walking period within the volume. Note that in the Figure 3.4 at right side, model window there are heel markers missed. Those markers removed since heel strike leads to motion artifact. They are changed to virtual markers in walking trials. Steps are same as procedures done in standing trail except more then one trial is considered.

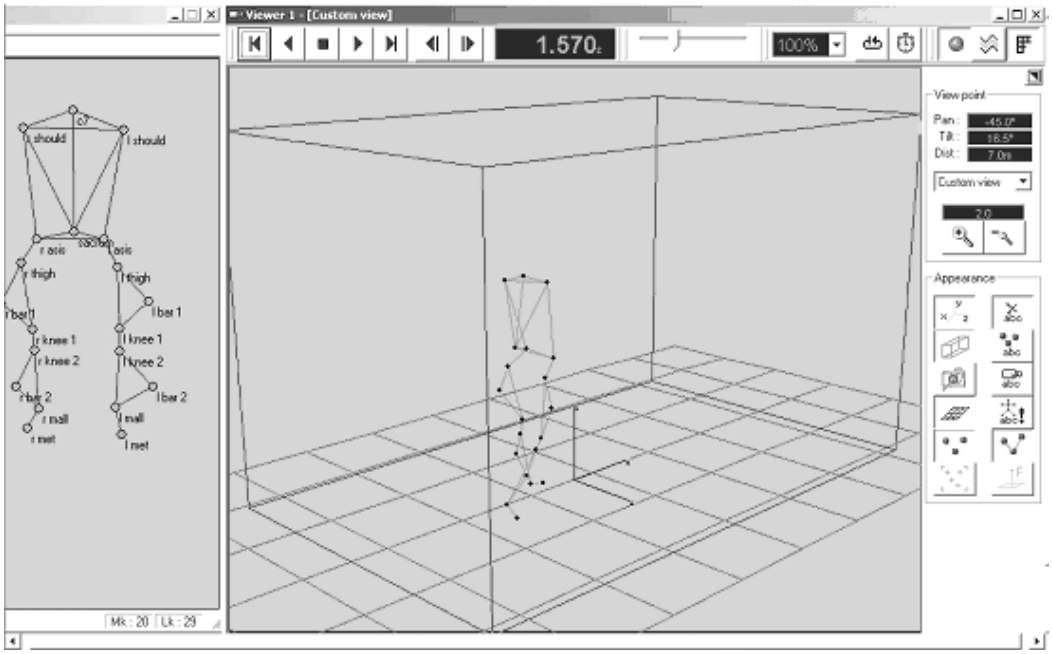


Figure 3.4 Tracking of the acquired walking trial.

- Elaboration of the acquired data

At the bottom left of Figure 3.5, step markers are placed for each left/right heel strike and toe off. This is necessary for defining the gait phases of subject. Elaboration of the post-processed data was necessary to define phases of the gait pattern. It is defined by marker placements at the time frames, which are placed at the moment of heel strike and toe off for both left and right foot. Physician who has good experience on step marker assignments should perform this procedure. It is necessary since all of the outputs are strictly dependent to the correctness of the assignment.

- Printing and exporting report of the subject to ASCII data

It is necessary that report taken should be in numerical data form for feeding Neural Network. Both form of the output is taken numerical data for simulation and print out for comparison of consistency with physical evaluation of the subject.

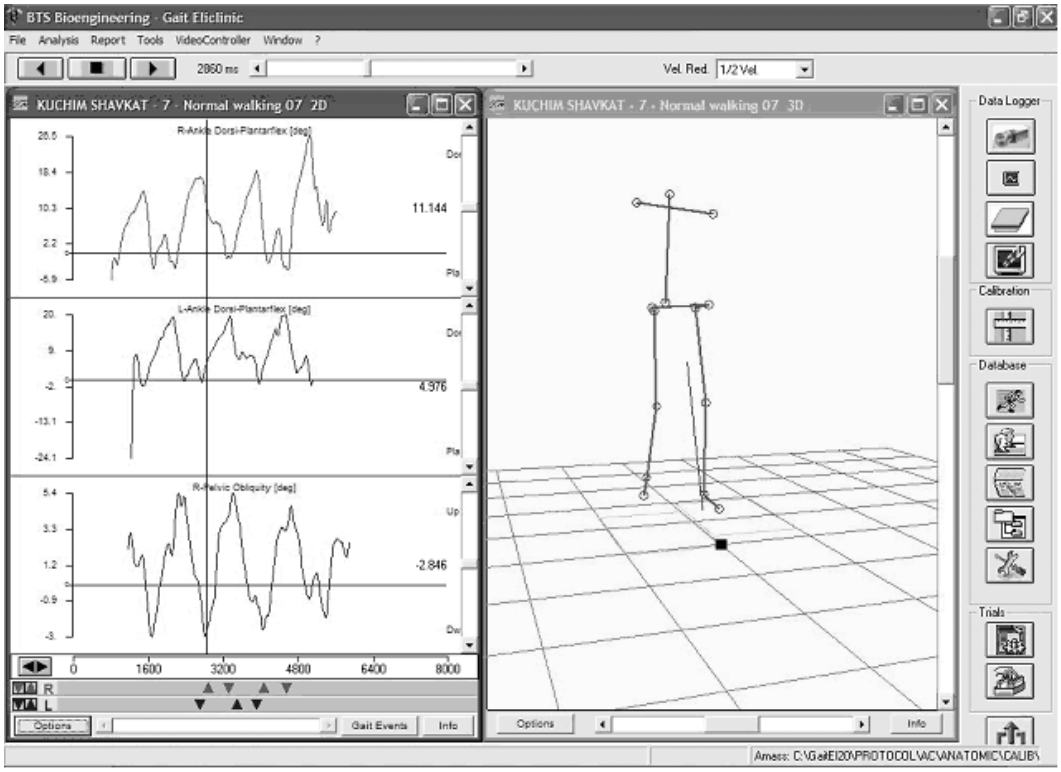


Figure 3.5 Elaboration of the processed walking trial.

3.2 Data Collection and Preprocessing

In this thesis our aim is to categorize the walking gait to normal and pathological, plus to classify later one if possible. Classifications are: normal gait, Right Hemiplegia, Left Hemiplegia and Spastic Diplegia. 187 of subjects were Spastic Diplegia, 58 of subjects were Right Hemiplegia, and 48 of subjects were Left Hemiplegia. Additionally, we have recorded 78 ‘healthy’ subjects. General characteristics of the selected subject illustrated as in tabular form below at Table 3.1.

Table 3.1 Characteristics of Selected Subjects (Age 3 –70).

GAIT CATEGORY	AGE		GENDER Female / Male	NUMBER OF CASES AVAILABLE
	Mean±S.D	Range		
Normal	23±12	6-60	30 / 48	177
Right Hemiplegia	15±13	5-58	22 / 36	200
Left Hemiplegia	19±18	4-70	19 / 26	142
Spastic Diplegia	13±5	3-23	60 / 127	226

Motion analysis trials were acquired more then once for each subject. Namely there are more then one cases for each subject. In the stage of processing of data Gait Eliclinic has algorithm that filters and interpolates saved data by Tracklab program. Kinematic trajectories were examined during trial elaboration stage. A little of the cases were ruined because of the damaged calibration, or have too much missing frames. Those trials with too much lost frames and calibration problem were abandoned. Report and ASCII of the elaborated trials obtained. Reported kinematic trajectories (kinematic graphs) were compared with physical evaluation measurements. Specifically, ankle and knee angles were taken into consideration. Irrelevant cases were removed. Finally, we were left with 177 normal walking, 226 Spastic Diplegia, 200 Right Hemiplegia and 142 Left Hemiplegia trials. Report and ASCII export of trials contains both kinematic and kinetic trajectories of subject. Since kinematics was used in this project, kinetic data were also abandoned. Within those kinematic outputs five pair of them were chosen. Those were chosen with supervision of medical exports of Motion Analysis Laboratory. Feature vectors are constructed from five pairs of kinematic trajectories, which is illustrated at Table 3.3. They are combined in consecutive order.

Acquired data revised again according to the age groups and motion analysis data consistency with video of the subject. Confinement of age between 6-12 left us with 24

Normal, 40 Right Hemiplegia, 31 Left Hemiplegia and 55 Spastic Diplegia subjects. For details of subjects see Table 3.2.

Table 3.2 General Characteristics of Subjects (Age 6 - 12).

GAIT CATEGORY	AGE		GENDER Female / Male	NUMBER OF CASES AVAILABLE
	Mean±S.D	Range		
Normal	9.8±2	6-12	12 / 12	110
Right Hemiplegia	8.8±1.8	6-12	19 / 21	134
Left Hemiplegia	8.7±2.2	6-12	13 / 18	114
Spastic Diplegia	9.3±1.8	6-12	25 / 55	160

After age confinement we left 150 subjects out of 371 recorded ones. Video consistency performed by comparing report of them with corresponding captured videos. Finally, we have extracted 518 useful trials after those procedures. We had 110 trials for Normal, 134 trials for Right Hemiplegia, 114 trials for Left Hemiplegia and 160 trials for Spastic Diplegia. Feature vectors constructed from bilateral kinematic trajectories explained in Table 3.3 for each case (trial).

Table 3.3 Gait Trajectories Utilized fot Classification.

Right-Ankle Dorsi-Plantarflex	Left-Ankle Dorsi-Plantarflex
Right-Hip Flex-Extension	Left-Hip Flex-Extension
Right-Knee Flex-Extension	Left-Knee Flex-Extension
Right-Pelvic Rotation	Left-Pelvic Rotation
Right-Pelvic Tilt	Left-Pelvic Tilt

Ten trajectories (five pair of kineatic ouput) are combined consecutively to form the feature vector Those ten trajectories are Ankle Dorsi-Plantar Flexion pair, Hip Flexion-Extension pair, Knee Flexion-Extension pair, Pelvic Rotation pair and Pelvic Tilt pair.

Figure 3.6 is one of those pairs as a graphical form that is generated report by BTS Elit program. Each of those curves contains 100 sample points. Thus obtained feature vectors have total of 1000 elements. Plot of the average graph for each classification group is available in Appendix B. Vertical axis is angle of knee flexion-extension in degree and horizontal axis is percent of the gait cycle. Graph in the Figure below contains bilateral output and normal curve for comparison. Curves in blue, red and black correspond to right side, left side and normal values of knee flexion extension in degrees. Shaded area around normal curve is deviation range of the normal patterns.

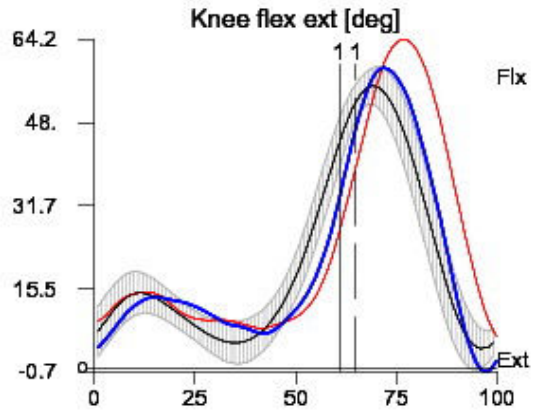


Figure 3.6 Knee Flex-Ext Graph, Sample Report.

Velocity and speed of the each subject may differ. In order to cope with that incompetence gait cycle is accepted as a standard instead of time, Figure 3.7. One complete period of walking is normalized to 100%. The gait cycle begins when one foot contacts the ground and ends when that foot contacts the ground again. Thus, each cycle begins at an initial contact with a stance phase and proceeds through a swing phase until the cycle ends with the limbs next to initial contact. The stance phase accounts for approximately 60 percent and the swing phase for approximately 40 percent of a single gait cycle. In the

Figure 3.7, it is shown that Gait cycle starts from toe off of the left leg and completes period with the same leg.

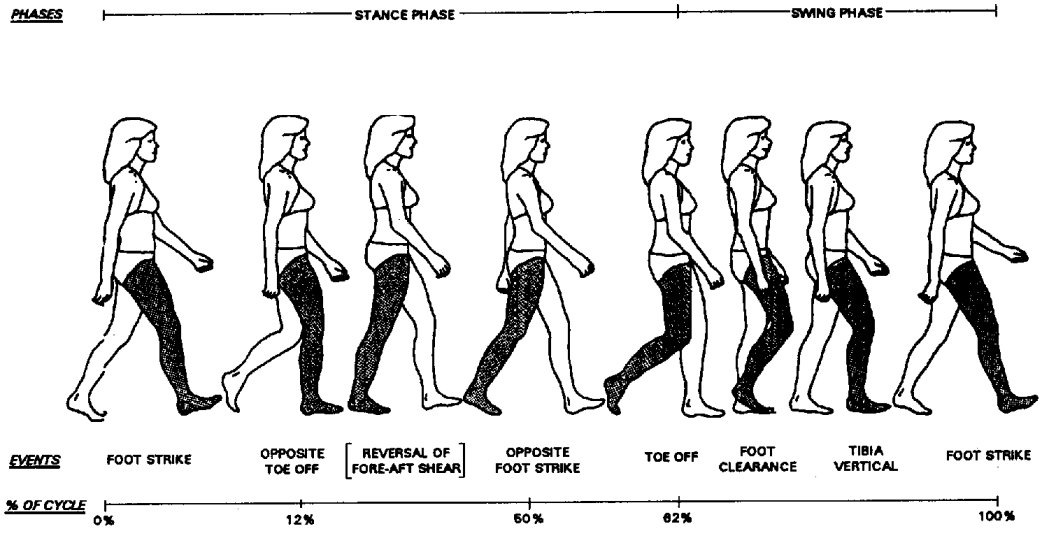


Figure 3.7 Gait cycle [7].

3.3 Neural Network Selection and Design

As the nature of the perceptron it is impossible to classify nonlinear inputs. However it is proven that linearly separable problems could be solved (i.e. OR problem). It has very simple algorithm and low computation cost but it could not be used in this project since complexity of the problem.

Secondly, we have unsupervised learning method that could be possibly used to this project. In unsupervised learning model there is no guiding still network with both inputs and outputs, no feedback from environment to dictate what these outputs should be. Architecture of the model is simple. Difference comes from learning rules. Most networks have one layer. Unsupervised learning needs data redundancy and it is prone to noise.

Finally, we have supervised learning model with multi-layer back-propagation algorithm. Although the greater power of multi-layer networks was realized long ago, it was

recently shown that how to make them learn a particular function [17]. It is successful to solve difficult and diverse problems and able to compensate noise.

Our goal in this study is to train Network and obtain reasonable results. We chose multilayer back propagation network since distinguishing four different groups is difficult problem and data used may have noise. Our success would be evaluated according to the results obtained in previous researches.

3.3.1 Simulating Back Propagation Multilayer Network

In this network model, the input units are fan-out processors only. That is, the units in the input layer perform no data conversion on the network input pattern. They simply act to hold the components of the input vector within the network structure. Thus, the training process begins when an externally provided input pattern is applied to the input layer of units [11,19]. Forward signal propagation then occurs according to the following sequence of activities:

1. Locate the first processing unit in the layer immediately above the current layer.
2. Set the current input total to zero.
3. Compute the product of the first input connection weight and the output from the transmitting unit.
4. Add that product to the cumulative total.
5. Repeat steps 3 and 4 for each input connection.
6. Compute the output value for this unit by applying the output function

   $f(x)= 1/(1 + \exp(x))$, where x - input total.
7. Repeat steps 2 through 6 for each unit in this layer.
8. Repeat steps 1 through 7 for each layer in the network.

Once an output value has been calculated for every unit in the network, the values computed for the units in the output layer are compared to the desired output pattern, element by element. At each output unit, an error value is calculated. These error terms are then fed back to all other units in the network structure through the following sequence of steps:

1. Locate the first processing unit in the layer immediately below the output layer.
2. Set the current error total to zero.
3. Compute the product of the first output connection weight and the error provided by the unit in the upper layer.
4. Add that product to the cumulative error.
5. Repeat steps 3 and 4 for each output connection.
6. Multiply the cumulative error by o, where o is the output value of the hidden layer unit produced during the feed forward operation.
7. Repeat steps 2 through 6 for each unit on this layer.
8. Repeat steps 1 through 7 for each layer.
9. Locate the first processing unit in the layer above the input layer.
10. Compute the weight change value for the first input connection to this unit by adding a fraction of the cumulative error at this unit to the input value to this unit.
11. Modify the weight change term by adding a momentum term equal to a fraction of the weight change value from the previous iteration.
12. Save the new weight change value as the old weight change value for this connection.
13. Change the connection weight by adding the new connection weight change
14. Repeat steps 10 through 13 for each input connection to this unit.
15. Repeat steps 10 through 14 for each unit in this layer.
16. Repeat steps 10 through 15 for each layer in the network.

3.3.2 Structure of the Simulated MLPBP and Input Data

Using algorithm in previous section and formulation of MLPBP in section 4.3.2 MATLAB code is written. Code saved in mlpbp.m file, as a standard MATLAB m file. Program trains a multilayer network by using back-propagation algorithm.

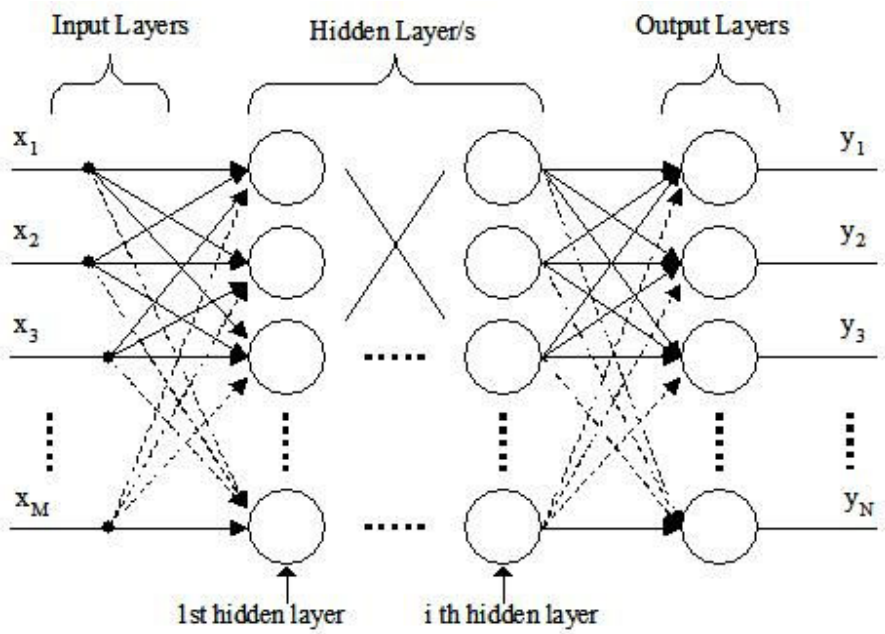


Figure 3.8 Neural Network Architecture.

Designed Neural Network has M many inputs. In the Figure 3.8 each input denoted by x_1, x_2, x_3,…,x_M. Since each kinematic trajectory collected has 100 sample points, subscript M is the multiple of 100. Thus each sample point of gait parameter is entered as an input. Input layer (Input node) has M = feature dimension. So entered file should have matrix M many input. Each sample point of the feature pattern fed to the network entered as an input.

Hidden layer/s. Defined hidden layer could be one or more then one, as well. Each time when program executed it asks number of hidden layer which means user could define the number of hidden layers in Network. Additionally, number of neurons supposed to be entered for each layer.

Output layer N = output dimension. Program defines number of the neurons to be deployed at the output layer. It is equal to the number of output and number of categories to be classified.

Designed ANN is very flexible. Most of the parameters can be specified by user: number of total layers, number of units in each layer, learning rate, momentum, error threshold, maximum number of epochs, number of iterations to stop when there is no improvement etc.

Parameters expected to enter:

- Number of output = N
- Number of total layers including output layer = L
- Number of neurons in each layers L1, L2,…Li
- Steepness parameter = lambda (For transfer function)
- Learning rate = alpha (or eta in literature)
- Momentum constant = mom
- Error threshold = Et
- Maximum number of epoch to run
- Training Samples = Kr
- Number of epoch between convergence check
- Maximum number of iterations if no improvement
- Number of iteration to be plotted for Error

All of the synaptic weights are initialized to very small random numbers. A sigmoid is chosen as the nonlinear function. Program runs training up to maximum epoch defined by user. It terminates the process in case of no improvements detected in the error. Additionally, it stops training when minimum defined error (Error threshold) attained. So there is three criteria of termination: Maximum Epoch, No progress in Error and achievement of minimum Error. Data entered to network should be ASCII file. Trainig file and Test file should be

entered separately. Files contain matrix data, which is shown in Figure3.9. Matrix contains K many feature vectors. There is M many inputs and N many target output in each feature vector. M correspond to the number of sample point extracted from ten kinematic trajectory. Target output that is last N many element of the feature vector consists of zeroes and one. The element that equals to one defines to which group it belongs. User defines Kr many feature vectors that would be used for training network out of entered K many samples. For estimation feature vector is picked randomly at each iteration from dataset with Kr many feature vectors. This procedure known as shuffling of the patterns and it is perfomed for better performance.

Two different codes are written Mlpbp.m and Mlpbp_cv.m. Programs expect as an input data in ASCII file. Two different data sets Training and Test file should be entered. Second code needs Cross-Validation as well as Training and Test data file.

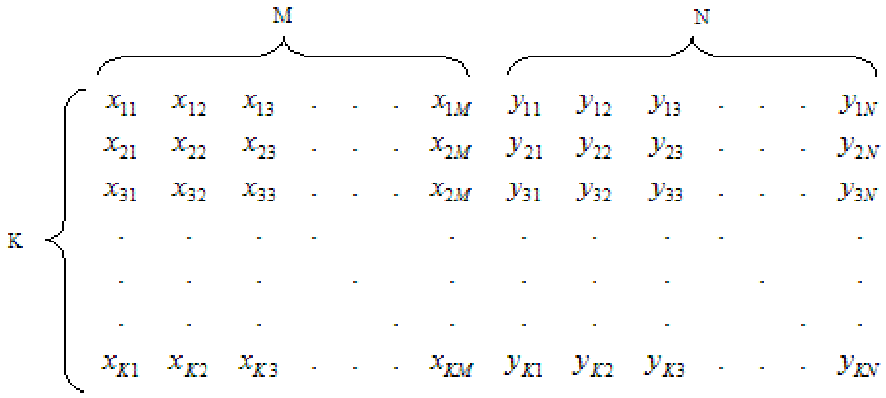


Figure 3.9 Data Structure of Input File.

3.4 Data Implementation

3.4.1 Neural Network Performance Verification with XOR

We know that single-layer perceptron cannot classify input patterns that are not linearly separable. However, nonlinearly separable patterns are of common occurrence. For example, this situation arises in the Exclusive OR (XOR) problem, which may be viewed as a special case of a more general problem, namely that of classifying points in the unit

hypercube. Each point in the hypercube is either in class 0 or class 1. However, in the special case of the XOR problem, we need consider only the four corners of the unit square that correspond to the input patterns A (0,0), B (0,1), C (1,0), and D (1,1). Input patterns and Truth Table for XOR function illustrated in tabulated form at Table 3.1.

Table 3.4 Truth Table of XOR function.

PATTERN	INPUT X_1	INPUT X_2	OUTPUT
A	0	0	0
B	0	1	1
C	1	0	1
D	1	1	0

Therefore XOR problem is implemented to check whether designed neural network could adapt to solve the problem. In the Figure 3.10 below it is obvious that there is no line that could separate two different class of output. Classes are not linearly separable.

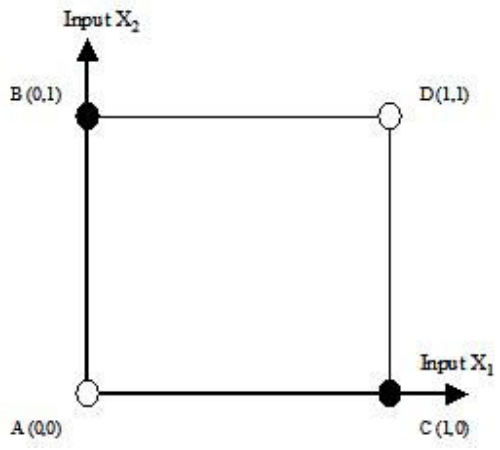


Figure 3.10 Geometrical Representation of XOR function.

Since written MATLAB code accepts as inputs ASCII file in the matrix form, two set of files generated for training and test. Matrices contain three column vectors first corresponds to input X_1, second to input X_2 and third to target output. Raw of the matrices are pattern samples, which consist of element of X_1, X_2 and target output.

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Figure 3.11 Train and Test matrices.

In the Figure 3.11 above first logic array is train input which is 4x3 matrix file and second one is test input, which is 4x3 matrix file. Note that third column is target output.

3.4.2 Network Training with Single Gait Parameter

In order to take reliable improvements, datasets applied with reduced dimensions. The more distinct differences are observed at Ankle Dorsi-Plantar Flexion graph, see Appendix B. In the sense of medical properties, it contains very important information. This kinematic data is chosen with advices of medical specialists.

Unilateral ankle trajectory is deployed for training the network in the first trial. It is trained to classify Normal and Diplegia. Entered input set contains sample points of one side of the Ankle Dorsi-Plantar Felxion. Therefore input dimension is equal to 100. Network design has one hidden layer and one output layer. Hidden layer contains 100 neurons and output layer has 2 neurons.

Secondly, code is executed to train network with 200 input nodes (200 inputs), 200 hidden neurons and 2 outputs. Sample points of Right and Left Ankle Dorsi-Plantar Felxion are fed. That is bilateral ankle trajectories are used for classifications. First program executed to categorize Right Hemiplegia and Left Hemiplegia. Then Normal and Spastic Diplegia classification are performed with the same network design but with increased number of sample patterns and smaller learning rate. Numbers of samples for the first execution were 64 Right Hemiplegia and 64 Left Hemiplegia. On the other hand later execution has 64 Normal and 192 Spastic Diplegia sample patterns. Learning rate entered were 0.02 and 0.002 respectively.

Three different gait condition Normal, Spastic Diplegia and Right Hemiplegia were tried to classify. Bilateral ankle trajectories are used for training again. Because of the fact that number of categories has been changed network's output neuron number changed from 2 to 3. All other network characteristics remained same 200 input dimension 200 hidden layer neurons and 0.002 learning rate.

To show that pairs of feature vectors (Ankle Dorsi-Flexion Angle for Right and Left) are not sufficient, network was trained for classification of four different gait patterns, which are Normal, Right Hemiplegia, Left Hemiplegia and Spastic Diplegia. It is performed though insufficiency of features was emerged in previous execution at classification of three different conditions. In that simulation all network parameters remained the same except for the output neuron that changed from 3 to 4.

3.4.3 Network Training with Additional Gait Parameters

New files are created for training and test datasets. In addition to data pairs of Right and Left Ankle Dorsi-Plantar Flexion nine other feature trajectories are added. Those are: Right-Left Hip Flex-Extension, Right-Left Knee Flex-Extension, Right-Left Pelvic Rotation and Right-Left Pelvic Tilt. Created file implemented to the ANN for classification of four different groups.

Program executed for training network with new data set. Architecture of it remained the same but learning constant assigned as 0.005. Owing to the obtained results, architecture of the network is decided to modify. Heuristics for making network perform implied that sharp convergence of the amount of neurons in consecutive layers leads to low success in categorization. This time three hidden layers selected. Network architecture has following design 1000 inputs at input node, 1000 neurons in the first hidden layer, 80 neurons in the second hidden layer, 8 neurons in the last hidden layer and four output neurons. As shown in the previous chapters this could be notated as follow: 1000-1000-80-8-4. Learning rate was 0.002. Another simulation with different network design 1000-1000-200-10 performed to investigate progress in network performance. All other parameters kept the same.

3.4.4 Effect of Learning Rate

Several simulations are accomplished with different learning constants. Which has incredible impact to both convergence and network performance. Both our past experience and literature approve that it has optimal range. Some of those simulations are added to manifest learning constant effect. Network design is 1000-1000-80-8-4 which give better results then other tried ones. ANN program executed five times with different learning constants; 0.2, 0.02, 0.002, 0.0002 and 0.00002.

3.4.5 ANN Training with Confined Data

Neural network is fed with dataset that collected from the subjects who are confined to 6 to 12 ages. Network fixed to the 1000-1000-80-8-4 architectural design. Learning constant is 0.002.

Next, age and sex parameter of the subjects added to the feature vector. Since extra two parameters added dimension of the input feature vector changed to 1002. Thus network architecture is as follows: 1002-1002-80-8-4 To check consistence of the network results several execution is needed with different dataset. Pool of data is created that contains feature vectors. Matlab program code is written that generates training and test file. It picks a feature vector randomly and generates two different files. Each feature vector is selected only once, namely generated files are completely different.

Program is ran 9 times to generate 9 different input file sets. All of the files are applied to the ANN. The most logical reference point in data analysis is the mean, notated as $\bar{x}$, the typical value for a set of data. Mean of confusion matrix and classification success calculated.

The variance of the results are calculated as:

$$\sigma^2 = \frac{\sum (x - \bar{x})^2}{n-1} \qquad (4.1)$$

where n is the number of trials performed, or total number of x values.

Standard deviation, σ, is simply square root of the variance.

3.4.6 Generalization test of ANN

Generalization test is tests of the NN's ability to classify samples correctly which are never seen before. In back-propagation learning, we typically start with a training sample and use the back-propagation algorithm to compute the synaptic weights of a multilayer perceptron by loading (encoding) as many of the training examples as possible into the network. The aim is that the neural network so designed will generalize. A network is said to generalize well when the input-output mapping computed by the network is correct (or nearly so) for test data never used in creating or training the network; the term "generalization" is borrowed from psychology [12].

A neural network that is designed to generalize well will produce a correct input-output mapping even when the input is slightly different from the examples used to train the network. When, however, a neural network learns too many input-output examples, the network may end up memorizing the training data. It may do so by finding a feature (due to noise, for example) that is present in the training data but not true of the underlying function that is to be modeled. Such a phenomenon is referred to as overfitting or overtraining. When the network is overtrained, it loses the ability to generalize between similar input-output patterns.

3.4.7 Cross Validation

The hope is that the network becomes well trained so that it learns enough about the past to generalize to the future. From such a perspective the learning process amounts to a choice of network parameterization for this data set. More specifically, we may view the network selection problem as choosing, within a set of candidate model structure (parameterizations), the "best" one according certain criterion.

Data pool collected is partitioned into three disjoint subsets:

- Training set
- Validation set
- Test set

Since we have design of large neural network, cross-validation should be deployed to have a good generalization. The motivation here is to validate the model on a data set different from the one used for parameter estimation and test.

Dataset is partitioned into 0.5, 0.25 and 0.25 ratios to generate estimation, cross-validation and test files. So 75 percent of the whole data is used for network training meanwhile 25 percent of it used to validate training progress. Remained 25 percent of data used to test final trained ANN. “Best” network design and learning rate attained are used. Matlab code is modified to deploy cross-validation and trace its error propagation. Since rate of the partitioned data is different, plotting total error of training and validation is meaningless. Therefore mean squared error is plotted instead of total error so that one could compare error curves.

3.4.8 Early Stopping Method

A multiplayer perceptron trained with the back-propagation algorithm learns in stages, moving from the realization of fairly simple to more complex mapping functions as the training session progresses. A typical situation the mean-squared error decreases with an increasing number of epochs during training: it starts off at a large value, decreases rapidly, and then continues to decrease slowly as the network makes its way to a local minimum on the error surface. With good generalization as the goal, it is very difficult to figure out when it is best to stop. It is possible for the network to end up overfitting the training data if the training session is not stopped at the right point.

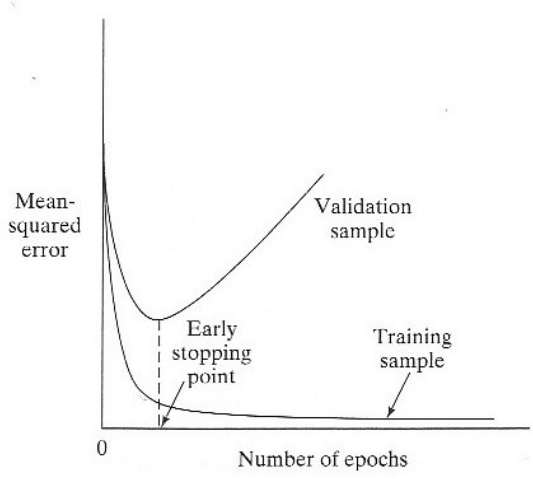


Figure 3.12 The Early-stopping rule based on cross-validation [12].

Figure 3.12, above, conceptualizes forms of two learning curves: one pertaining to measurements on the estimation subset and the other pertaining to the validation subset. Network simulated and training terminated by means of Early Stopping Method of Training.

3.4.9 Classification of Normal and Gait Disorders

Finally, a network is trained with revised data to distinguish Healthy gait pattern and Pathological one. It should be noted that Right Hemiplegia, Left Hemiplegia and Diplegia feature vectors used as pathological gait patterns. Classification of “Healthy” and Pathological is accomplished. Neural Network architecture remained the same except output layer neuron number. Since we have only two groups to classify, number of neurons at output layer is 2.

4. RESULTS

Average of feature vectors are plotted for each gait groups such that one would have an idea about their trajectories.

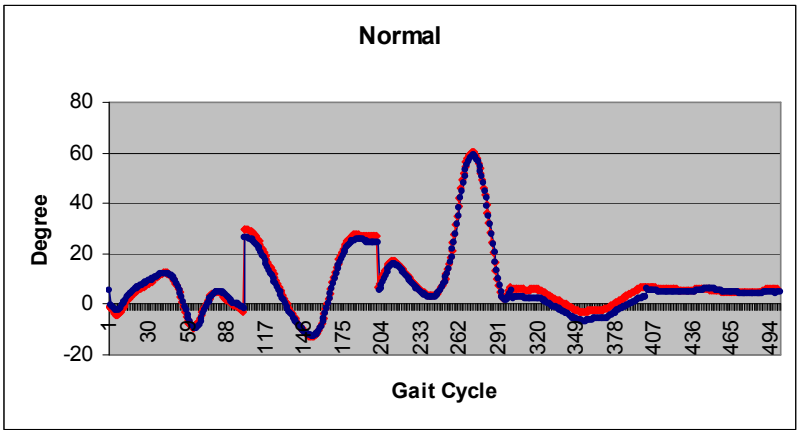


Figure 4.1 Average graph of normal sample patterns

Average graph of all 'healthy' subjects is depicted in Figure 4.1. Note that both curves for the right and for the left side are aligned. They almost fall on top of each other. Blue curve corresponds to the trajectories of the right and red one to the trajectories of the left side.

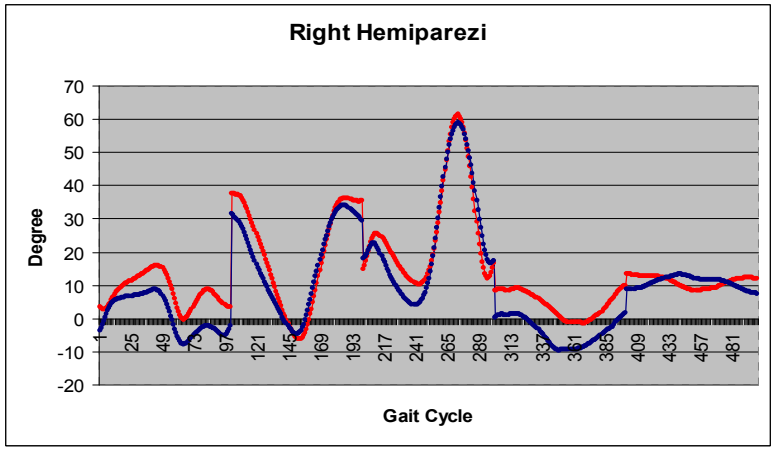


Figure 4.2 Average graph of Right Hemiplegia sample patterns

Average graph of all Right Hemiplegia subjects do not align for the right and the left sides. The result is as expected since one side of the body is affected more than other in Hemiplegia patients. Right limbs are more affected in Right Hemiplegia patients. In Figure 4.2 blue trajectory corresponds to the average of feature vectors of the right side and red one to the left side.

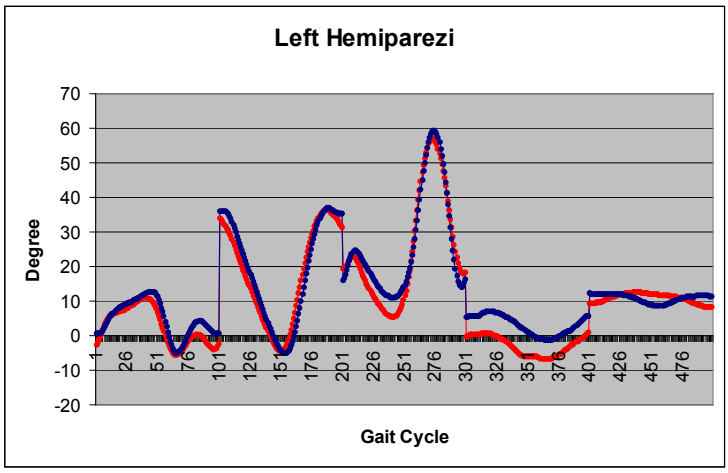


Figure 4.3 Average graph of Left Hemiplegia sample patterns

Here we have previous scenario again in Figure 4.3; average graph of kinematics data of Left Hemiplegia disorders do not align for right and left side. Which is not symmetric due to their unilateral affected limbs. Blue and red curves correspond to the average of the feature vectors of the right and left side respectively.

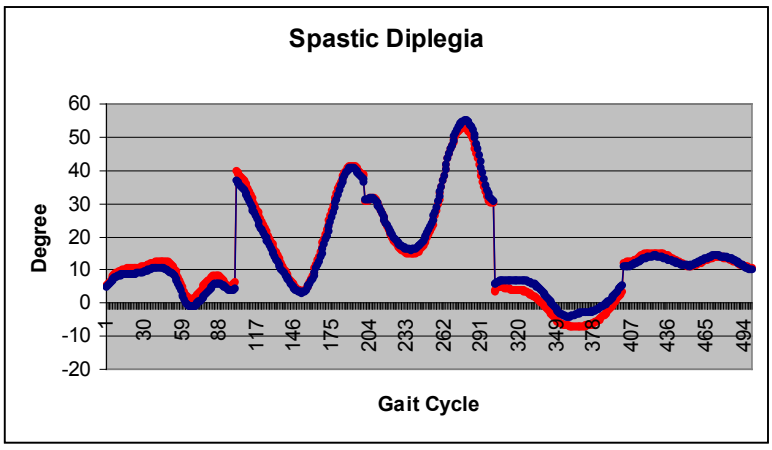


Figure 4.4 Average graph of Spastic Diplegia sample patterns

Average graph of Diplegic subjects have most curves aligned except for pelvic rotation. Graph in Figure 4.4 approves the fact that reviewed at section 4.2. All limbs are affected but lower extremity is affected more then upper ones.

4.1 Results of Implemented XOR problem

It is almost a standard procedure to apply XOR as a test of nonlinear problem solving capability of a Neural Network.

Table 4.1 Confusion Matrix and Classification Achievement for XOR problem.

	TRAINING		TEST	
Estimated / Real	First Group	Second Group	First Group	Second Group
First Group	2	0	2	0
Second Group	0	2	0	2
Clasification Success in %	100		100	

Table 4.1 is tabulated form of the obtained training results. There are confusion matrices, plus achieved classification rate in percentage for both training and test datasets. Two by two table in light green color is the confusion matrix for the training.

Table in dark color is the confusion matrix for the test. Cells under them correspond to the success attained for them. After that we always imply to one of those tables when we use confusion matrix term. Columns of the matrix contain number of feature vectors that estimated as a corresponding group. Rows of the matrix imply to the number of samples that is really belongs to that group.

From the Table 4.1, it is obvious that both training and test classifications are successful.

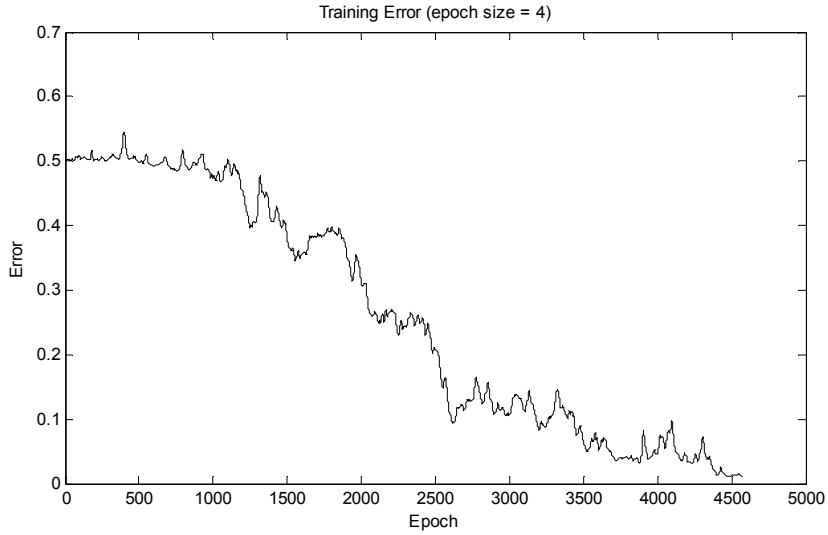


Figure 4.5 Total Training Error vs. # of epochs for the XOR problem.

Total error curve also converged to the threshold value, which was 0.01 in Figure 4.5. We know that ANN designed could classify nonlinear problems. Now we can implement our dataset to the program.

4.2 Results Network Implementation with Single Gait Parameter

It is proven that ANN could classify nonlinear problems. In order to take reliable improvements, datasets applied with reduced dimensions.

Table 4.2 Confusion Matrix and Classification Success for Normal and Spastic Diplegia.

	TRAINING		TEST	
Estimated / Real	Normal	Diplegia	Normal	Diplegia
Normal	50	0	42	8
Diplegia	0	50	6	44
Clasification Success in %	100		86	

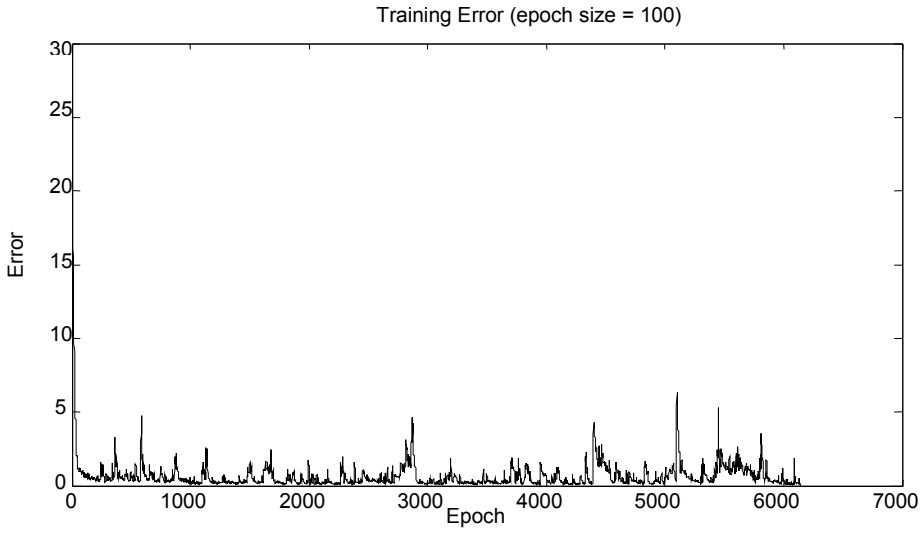


Figure 4.6 TTE vs. # of epochs for classification of Normal and Spastic Diplegia.

Classification success for the training and test are 100% and 86%, respectively. Diagonal elements of confusion matrix are the number of correct classified sets. First element of the confusion matrix corresponds to Normal and last element of the confusion matrix corresponds to Spastic Diplegia. Namely, 42 samples of normal set out of 50 and 44 samples of Spastic Diplegia out of 50 were sorted correctly.

Secondly, Right Hemiplegia and Left Hemiplegia data sets were applied to the ANN to observe whether network was able to distinguish between those similar characteristic disorders.

Table 4.3 Confusion Matrix and Classification Success for Right and Left Hemiplegia.

	TRAINING		TEST	
Estimated / Real	Right Hemiplegia	Left Hemiplegia	Right Hemiplegia	Left Hemiplegia
Right Hemiplegia	64	0	52	12
Left Hemiplegia	0	64	22	42
Clasification Success in %	100		73.4	

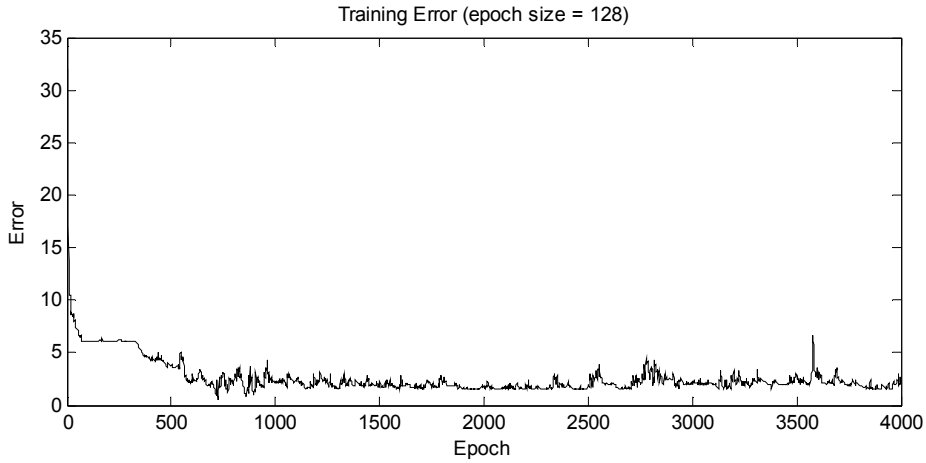


Figure 4.7 TTE vs # of epochs for classification of Right and Left Hemiplegia.

Table 4.3 is the result for training and test data of Right and Left Hemiplegia. First element of the confusion matrix corresponds to Right Hemiplegia and the last one to Left Hemiplegia.

Simulation is repeated for the first simulation. Gait types are Normal and Diplegia. Now we have data set of Ankle Dorsi-Plantar Flexion with both sides included.

Table 4.4 Simulation results for Normal and Spastic Diplegia with η=0.02.

	TRAINING		TEST	
Estimated / Real	Normal	Diplegia	Normal	Diplegia
Normal	64	0	44	20
Diplegia	0	192	9	183
Clasification Success in %	100		88.7	

There are 64 feature vector samples for normal and 192 feature vector samples for Diplegia. All of the training samples are estimated correctly. However 44 samples of normal gait and remained samples estimated as Diplegia. 9 samples of Diplegia confused as normal gait. Comparison of first and current simulation show that increasing number of samples and decreasing learning rate leads to increase in success of classification from 73.4% to 88.7%

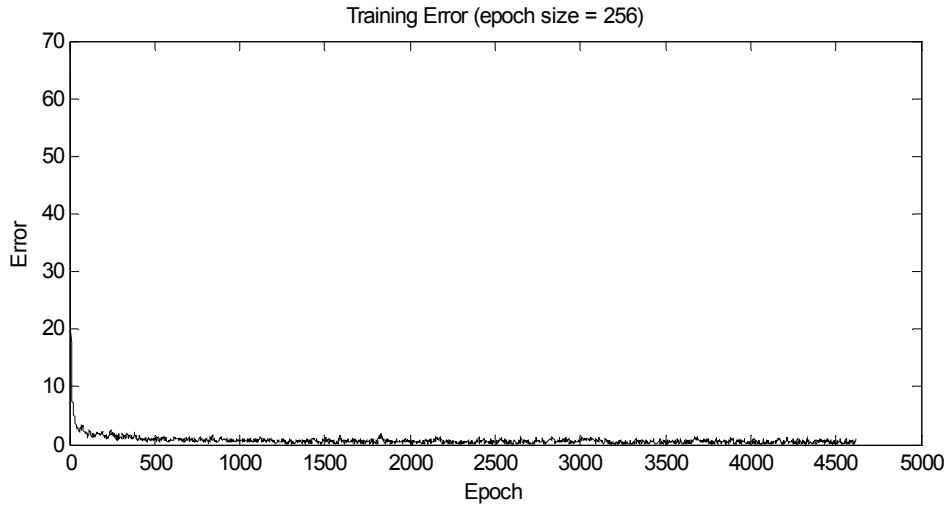


Figure 4.8 TTE vs. # of epochs for classification of Normal and Diplegia with η=0.02.

We again deploy one pair of feature vector of kinematic data, which is Ankle Dorsi-Plantar Flexion. Up to now it was successful to classify two groups. Would it be enough for classification of more than two groups? In order to get the answer further simulation was indispensable. Therefore three different types of gait (Normal, Spastic Diplegia and Right Hemiplegia) are applied to Network.

Table 4.5 Simulation results for Normal and Spastic Diplegia and Right Hemiplegia.

	TRAINING			TEST		
Estimated / Real	Normal	Diplegia	Right Hemiplegia	Normal	Diplegia	Right Hemiplegia
Normal	64	0	0	51	7	6
Diplegia	0	64	0	3	37	24
Right Hemiplegia	0	0	64	6	15	43
Clasification Success in %	100			68.2		

All of the samples in training set classified correctly. On the other hand percentage of success is decreased in test since pair of feature vector provided does not carry adequate information for good classification.

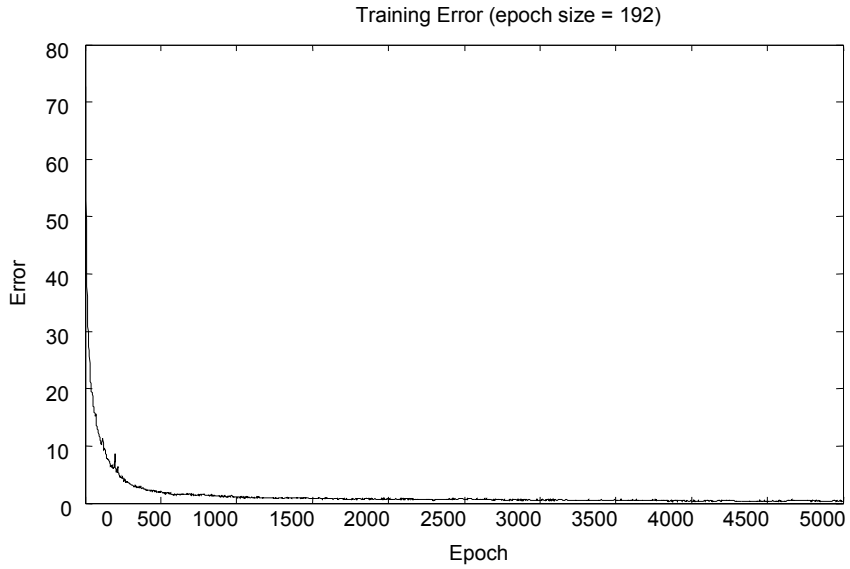


Figure 4.9 TTE vs. # of epochs for classification of Normal, Diplegia and Right Hemiplegia.

To show inefficiency of pairs of feature vectors which are Right and Left Ankle Dorsi-Plantar Flexion, another simulation is done. Inputs of four different groups are entered to check classification success. Those are Normal, Spastic Diplegia, Right and Left Hemiplegia respectively.

Table 4.6 Simulation results for 4 different gait conditions.

	TRAINING				TEST			
Estimated / Real	Normal	Diplegia	Right Hemiplegia	Left Hemiplegia	Normal	Diplegia	Right Hemiplegia	Left Hemiplegia
Normal	64	0	0	0	44	9	8	3
Diplegia	0	64	0	0	1	18	26	19
Right Hemiplegia	0	0	64	0	5	10	35	14
Left Hemiplegia	0	0	0	64	12	10	17	25
Clasification Success in %	100				47.7			

As expected, classification is rapidly decreased to 47.7%, which is the worst simulation result ever taken. Hypothesis of insufficient feature vector is proven.

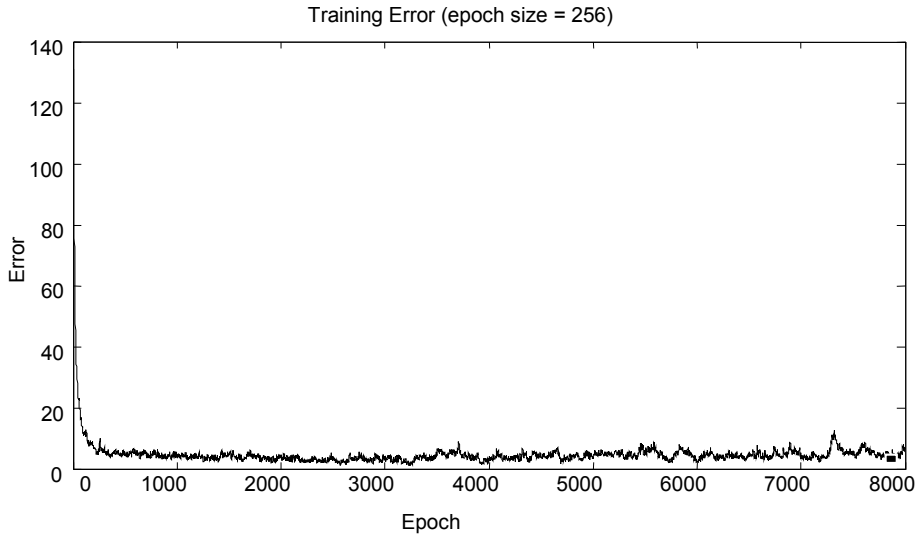


Figure 4.10 TTE for classification of 4 different gait types with Ankle pattern pair.

Simulations so far show that classification success of more then two groups is not adequate using only pair of Ankle trajectories. As the number of categorization increases need of additional feature is increased. We concluded that other pairs of feature vectors should be utilized for good performance of Network.

4.3 Results of Classification with Additional Gait Parameters

Added gait parameter pairs are following; Right-Left Ankle Dorsi-Plantar Flexion, Right-Left Hip Flex-Extension, Right-Left Knee Flex-Extension, Right-Left Pelvic Rotation and Right-Left Pelvic Tilt. Created file implemented to the ANN for classification of four different groups.

Despite of that training does not completed successfully, attained results for test is better then previous simulation. It is shown that training success is remained at 67.8 % and attained test result is 56.8% at Table 4.7. Only 2 actual sample of normal gait is correctly estimated which may be consequence of incomplete process of training.

Table 4.7 Confussion Matrix and Classification Success for 4 different gait type.

	TRAINING				TEST			
Estimated / Real	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia
Normal	2	18	53	27	1	17	47	35
Right Hemiplegia	0	87	4	9	0	66	10	26
Left Hemiplegia	0	7	89	4	0	10	77	13
Diplegia	0	3	4	93	0	7	10	83
Clasification Success in %	67.8				56.8			

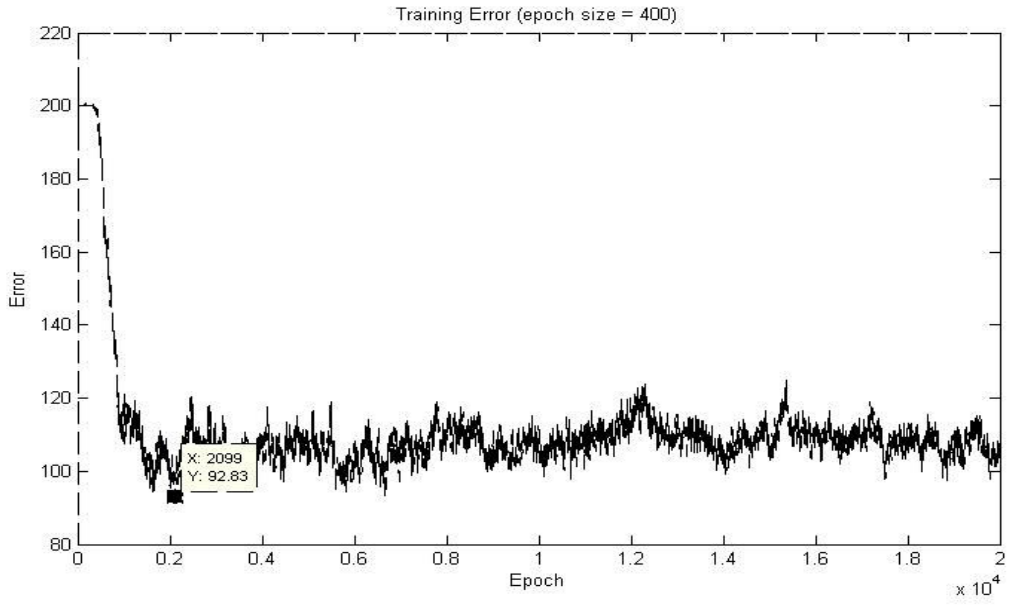


Figure 4.11 TTE for classification of four gait types with all collected data.

Increasing feature of the inputs increased performance of the ANN in the previous simulation. However result is far from being a good success. Sharp convergence of the amount of neurons in ANN may lead to low success in categorization. Thus network architecture is modified from 1000-1000-4 to 1000-1000-80-8-4. Program executed for training the Neural Network with modified architecture.

Table 4.8 Confussion Matrix and Classification Success for 4 different gait type.

	TRAINING				TEST			
Estimated / Real	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia
Normal	64	0	0	0	58	0	3	3
Right Hemiplegia	0	64	0	0	1	54	7	2
Left Hemiplegia	0	0	64	0	3	17	43	1
Diplegia	0	0	0	64	1	23	14	26
Clasification Success in %	100				70.7			

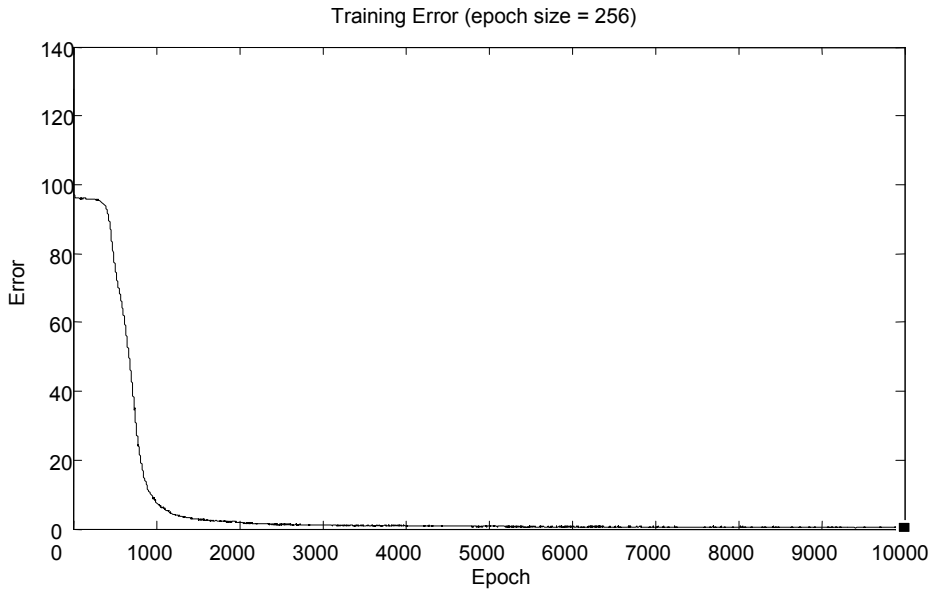


Figure 4.12 TTE for classification of Normal, Diplegia, Right and Left Hemiplegia.

Results obtained were great. We have 100 percentage of correct estimation in training samples. Rate of successful classification is very good. It is 32.2 % more for training and 13.9 % more for test than previous results. Looking at diagonal elements of confusion matrix of test, we conclude that the number of correctly estimated samples increase at great amount, see Table 4.8

Architectural design is modified for further improvements.

Table 4.9 Confussion Matrix and Classification Success for 4 different gait type.

	TRAINING				TEST			
Estimated / Real	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia
Normal	64	0	0	0	57	0	6	1
Right Hemiplegia	0	64	0	0	0	50	7	7
Left Hemiplegia	0	0	64	0	17	11	32	4
Diplegia	0	0	0	64	18	5	4	37
Clasification Success in %	100				68.8			

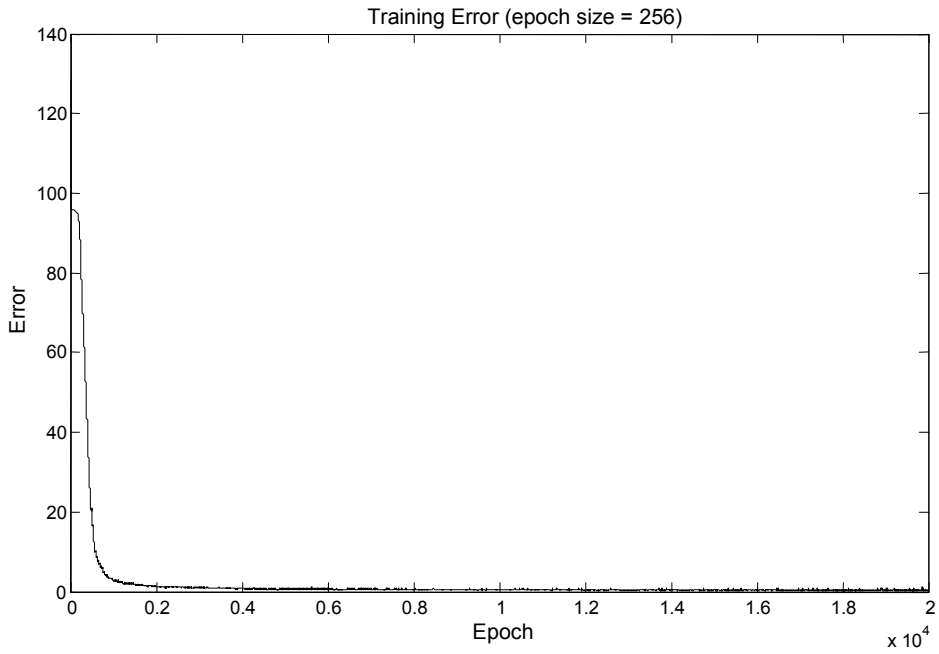


Figure 4.13 TTE of classification with 1000 – 1000 – 200 – 20 ANN design.

From the results above, it is obvious that there is optimum amount of numbers of neurons that should be used in ANN. Nonetheless; all of the parameters are remained the same except for the amount of neurons at the hidden layers, classification success of the trained network decreased.

4.4 Effect of Learning Rate constant

Several simulations are accomplished with different learning constants: 0.2, 0.02, 0.0002 and 0.00002. Some of those simulations are accomplished to manifest learning constant effect. Our best result obtained when learning rate η is equal to 0.002.

Table 4.10 Confusion Matrix and Classification Success for 4 different gait type

	TRAINING				TEST			
Estimated / Real	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia
Normal	0	51	0	4	0	26	0	1
Right Hemiplegia	0	45	0	22	0	15	0	18
Left Hemiplegia	0	22	0	35	0	10	0	18
Diplegia	0	0	0	80	0	0	0	40
Clasification Success in %	48.3				43			

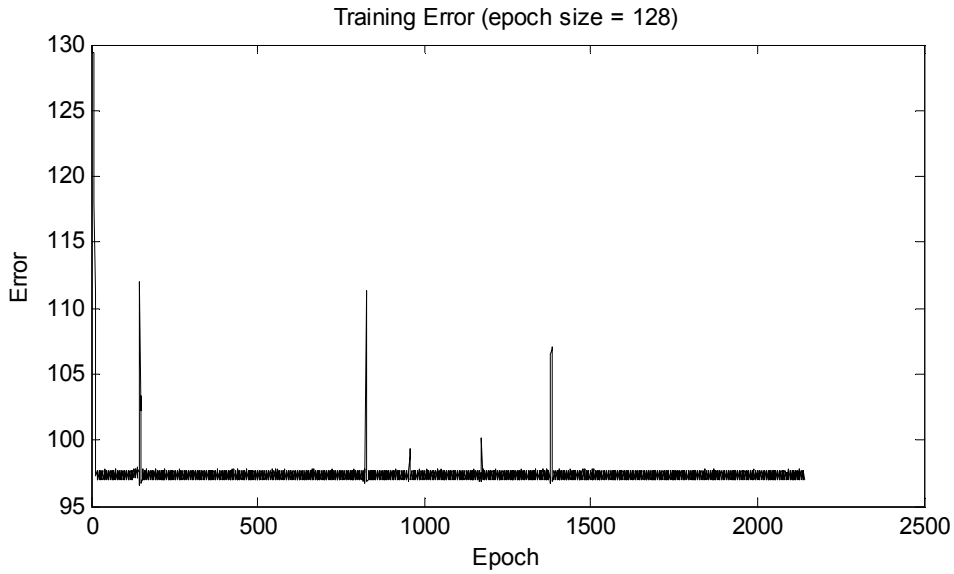


Figure 4.14 Total Training Error for classification with η =0.2.

Outcomes of the simulation above are inadequate. Even all iterations are not performed and training terminated since no improvement is achieved for 1001 consecutive checks. It shows that learning rate chosen is large to obtain desired outcome. It is meaningful that error in the Figure 4.14 oscillates; supposed reason of it may be local minima.

Learning rate now set to $\eta = 0.02$. All input parameters are the same: neural architecture, maximum number of epoch and momentum.

Table 4.11 Simulation results for Network Training with Learning Rate η=0.02.

	TRAINING				TEST			
Estimated / Real	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia
Normal	64	0	0	0	44	1	9	10
Right Hemiplegia	0	63	0	1	0	53	10	1
Left Hemiplegia	0	2	62	0	10	7	45	2
Diplegia	0	1	2	61	14	10	7	33
Clasification Success in %	97.7				68.4			

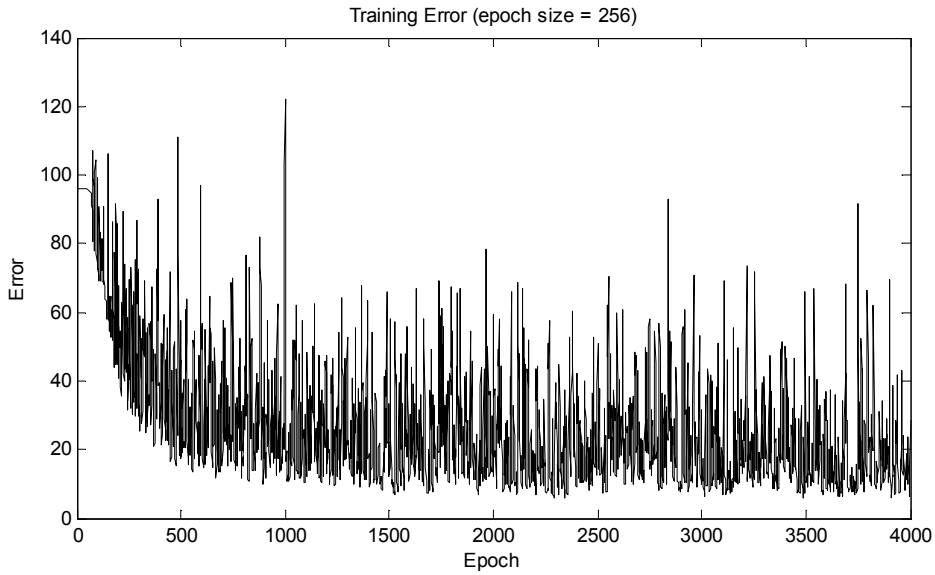


Figure 4.15 Total Training Error for classification with η=0.02.

From the figure above it is obvious that total Error of the simulation when η = 0.02 has too much fluctuations. And total Error does not lower then 10 units.

Learning rate is now set to η = 0.0002, which is smaller than estimated optimal learning rate.

Table 4.12 Simulation results for 4 Different Gait Type.

	TRAINING				TEST			
Estimated / Real	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia
Normal	63	0	0	1	57	0	7	0
Right Hemiplegia	1	54	5	4	1	48	12	3
Left Hemiplegia	5	8	50	1	29	4	26	5
Diplegia	6	24	8	26	16	29	9	10
Clasification Success in %	75.4				55.1			

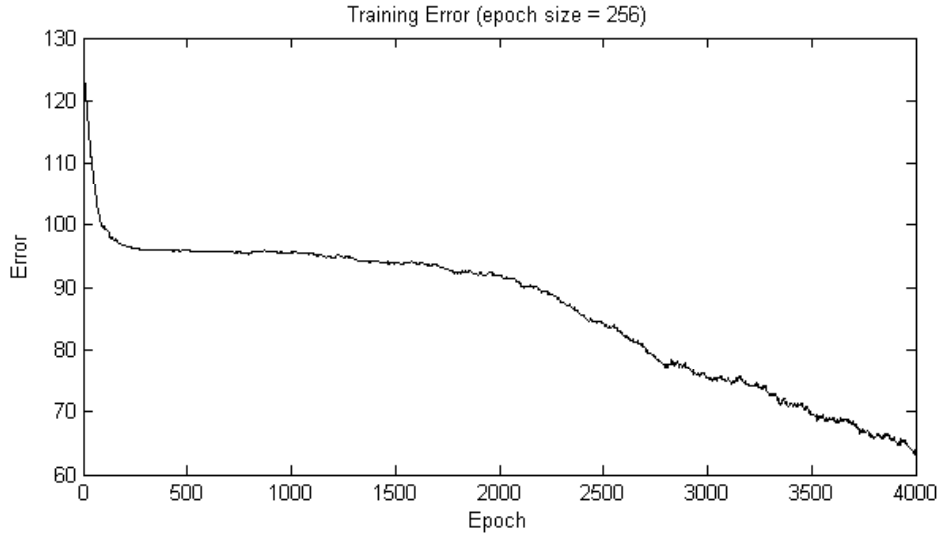


Figure 4.16 Total Training Error for classification with η =0.0002.

Expected scenarios are in Table 4.12 and in Figure 4.16. Although network has smooth progress, it has difficulty to converge at the same number of iterations.

Learning rate is now set to $\eta = 0.00002$, which is too small. It is exaggerated to show explicitly effect of the η to the simulation. All other input parameters are the same.

Table 4.13 Simulation results for 4 Gait Type with Learning Rate 0.00002.

	TRAINING				TEST			
Estimated / Real	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia
Normal	23	0	0	32	10	0	0	17
Right Hemiplegia	25	0	0	42	11	0	0	22
Left Hemiplegia	12	0	0	45	5	0	0	23
Diplegia	6	0	0	74	3	0	0	37
Clasification Success in %	37.5				36.7			

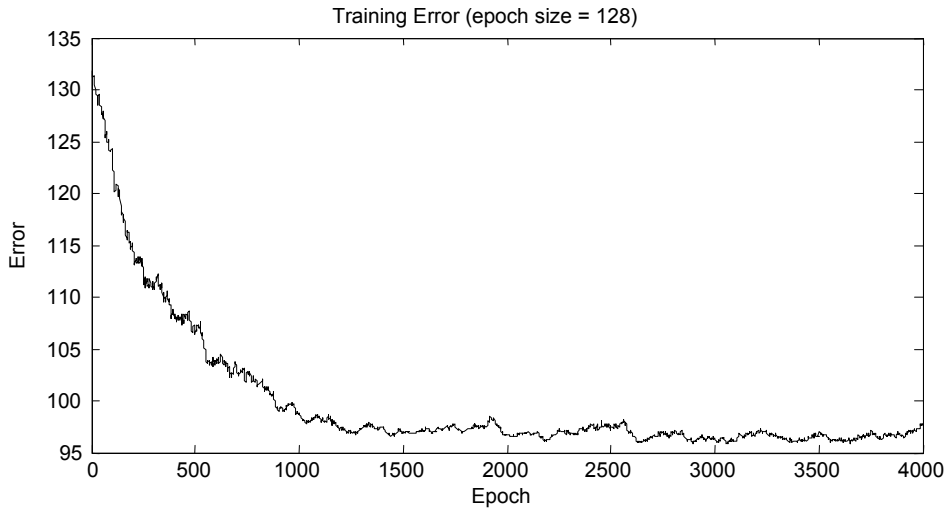


Figure 4.17 Total Training Error for classification with η=0.00002.

We had simulation that converges too slowly. It is obvious that total error does not get lower than 95, which was 65 at 4000'th iteration in previous one. It shows that small learning rate parameter results in slower convergence and it is a waste of time and resources.

In order to confirm our best result, simulation is repeated for learning rate η = 0.002 with all other parameters remaining the same.

Table 4.14 Confussion Matrix and Classification Success.

	TRAINING				TEST			
Estimated / Real	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia
Normal	64	0	0	0	55	0	6	3
Right Hemiplegia	0	64	0	0	0	42	7	15
Left Hemiplegia	0	0	64	0	12	1	40	11
Diplegia	0	0	0	64	10	4	5	45
Clasification Success %	100				71.1			

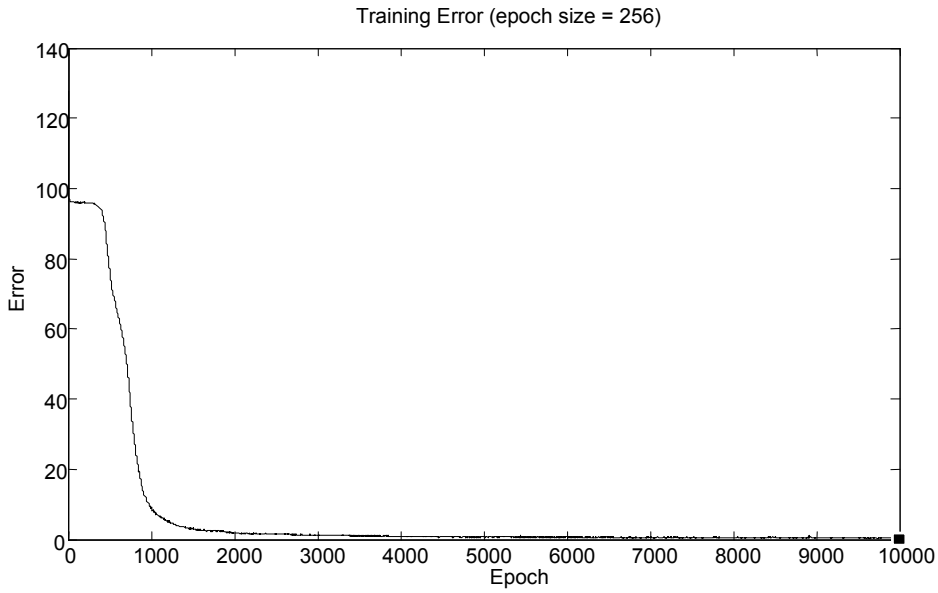


Figure 4.18 Total Training Error for classification with η =0.002.

Results are very consistent with outcome of previous best simulation, which was 0.002. Classification for previous one was 70.7% and for the current one is 71.1%. We conclude that learning rate close to one helps network to converge fast as it is obvious from simulations of that section. On the other hand small learning rate leads to smooth training and enables network to obtain “deeper” minima. One should be careful to choose learning constant that balances networks’ performance and speed.

4.5 Implementation of Revised Data

Network simulation repeated with the "best" parameters obtained. Three hidden layers deployed with 1000 neurons for first layer, 80 neurons for second hidden layer, and 8 neurons for third hidden layer. Learning rate is chosen 0.002. ANN is fed with new data.

Table 4.15 Simulation results for Normal, Diplegia, Right and Left Hemiplegia.

	TRAINING				TEST			
Estimated / Real	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia
Normal	55	0	0	0	52	2	1	0
Right Hemiplegia	0	67	0	0	6	52	0	9
Left Hemiplegia	0	0	57	0	12	4	31	10
Diplegia	0	0	0	80	0	6	5	69
Clasification Success in %	100				78.8			

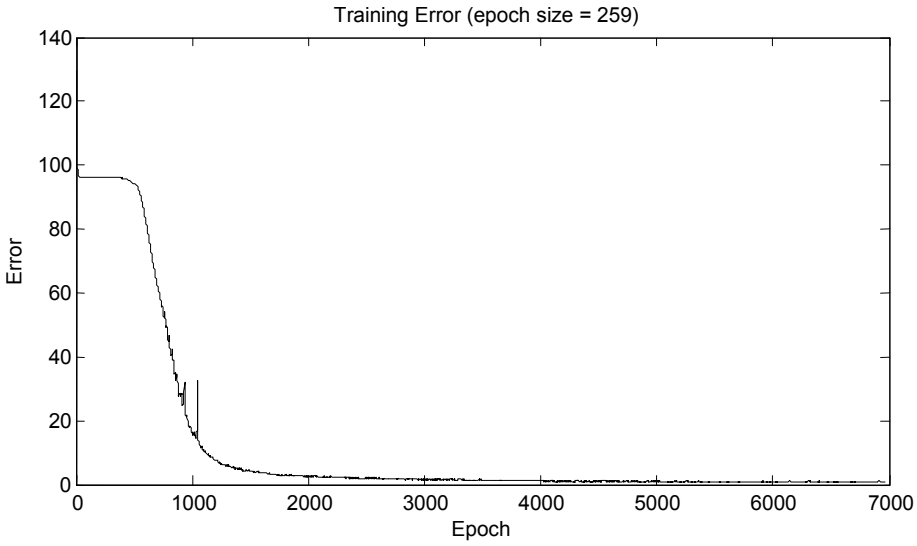


Figure 4.19 Total Error for classification after data revision.

Significant improvements are obtained. Approximately 7% of increase obtained in classification. Next, Sex and Age parameters are added to the input data. It shows that first rule of thumb is to have good data.

Table 4.16 Simulation Results with Sex and Age Parameters Included.

	TRAINING				TEST			
Estimated / Real	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia
Normal	55	0	0	0	46	7	2	0
Right Hemiplegia	0	67	0	0	4	54	5	4
Left Hemiplegia	0	0	57	0	13	2	39	3
Diplegia	0	0	0	80	0	9	7	64
Clasification Success in %	100				78.4			

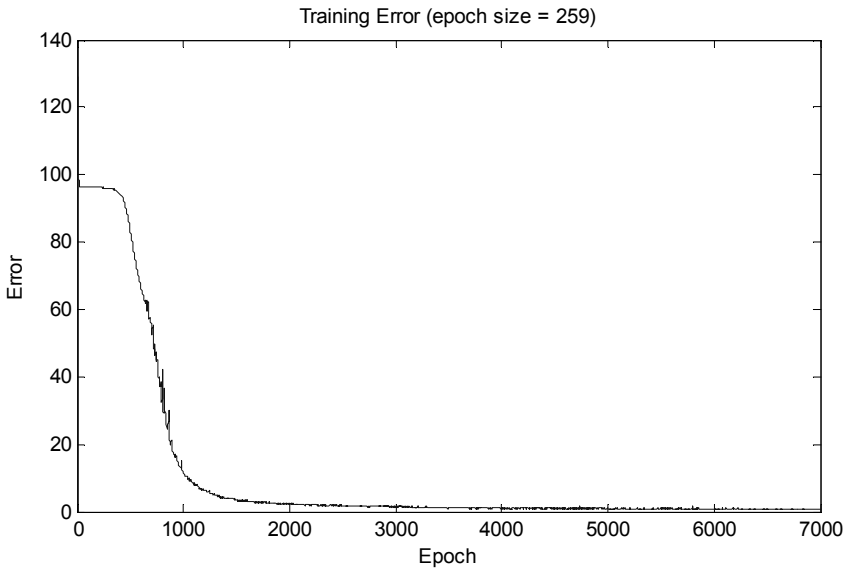


Figure 4.20 Total Error for classification with Sex and Age parameters.

Including Sex and Age parameters does not change the results significantly.

Simulation reliability is next issue supposed to be accomplished. Simulation is ran 9 times with 9 different input file sets. Results of applied files are available at Appendix C. Mean of results calculated both for confusion matrix and classification success demonstrated at Table 4.17.

Table 4.17 Average of nine simulation results.

	TRAINING				TEST			
Estimated / Real	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia
Normal	55	0	0	0	46	7	2	0
Right Hemiplegia	0	67	0	0	4	54	5	4
Left Hemiplegia	0	0	57	0	13	2	39	3
Diplegia	0	0	0	80	0	9	7	64
Clasification Success in %	100				77.6			

Classification results are [76.8 78.4 78.4 79.5 79.2 78.4 76.1 74.1 78]

Mean of results $\bar{x}$ = 77.6

The variance is calculated according to the formulation that mentioned in the subsection 3.3.7, $\sigma^2 = 2.9$.

Standard deviation is square root of the variance $\sigma =$ 1.7.

So obtained Results does not vary to much meaning they are consistent.

4.5.1 Results of Training with Cross Validation Data Set.

Finally we decided to modify our network for better generalization ability. According to the explained method in subsection 3.4.1 network modified and data generated to implement Cross-Validation .

Table 4.18 Simulation results with Cross Validation.

	TRAINING				TEST			
Estimated / Real	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia
Normal	55	0	0	0	22	1	4	0
Right Hemiplegia	0	67	0	0	0	25	2	6
Left Hemiplegia	0	0	57	0	2	3	18	5
Diplegia	0	0	0	80	0	2	1	37
Clasification Success in %	100				79.7			

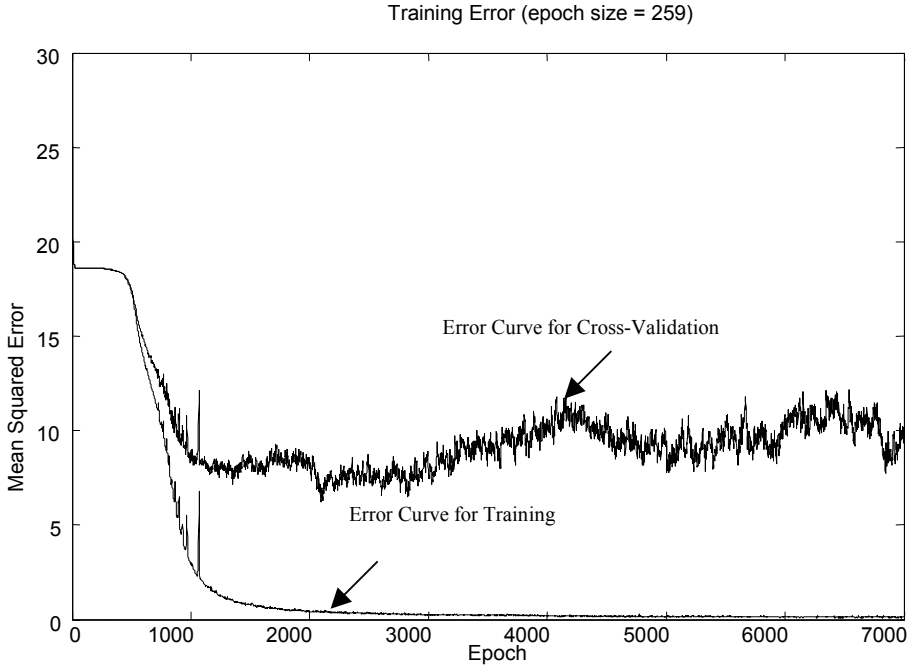


Figure 4.21 MSE of Training and Validation Data Sets.

In figure 4.20 above there are plotted Mean Squared Error curves for Cross Validation data set and Estimation data set. It is obvious from picture that as training proceeds Estimation Error asymptotically converges to small value. However Cross-Validation Error does not

decrease after a while. That implies to no improvements in generalization ability of the network even Error curve increases after that point.

4.5.2 Results of Early Stopped Training

Heuristically, now our NN should attain better result. All of parameters of the ANN are the same only simulation would be stopped by early stopping rule. According to Mean squared error of our simulation we chose stopping point as 1800. Table 4.19 shows result of the NN with same inputs fed and Early Stopping Epoch is 1800.

Table 4.19 Confusion Matrix and Classification Success After Early Stop of Training.

	TRAINING				TEST			
Estimated / Real	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia	Normal	Right Hemiplegia	Left Hemiplegia	Diplegia
Normal	55	0	0	0	21	2	4	0
Right Hemiplegia	0	67	0	0	0	27	0	6
Left Hemiplegia	0	0	57	0	0	2	24	2
Diplegia	0	0	0	80	0	1	2	37
Clasification Success in %	100				85.2			

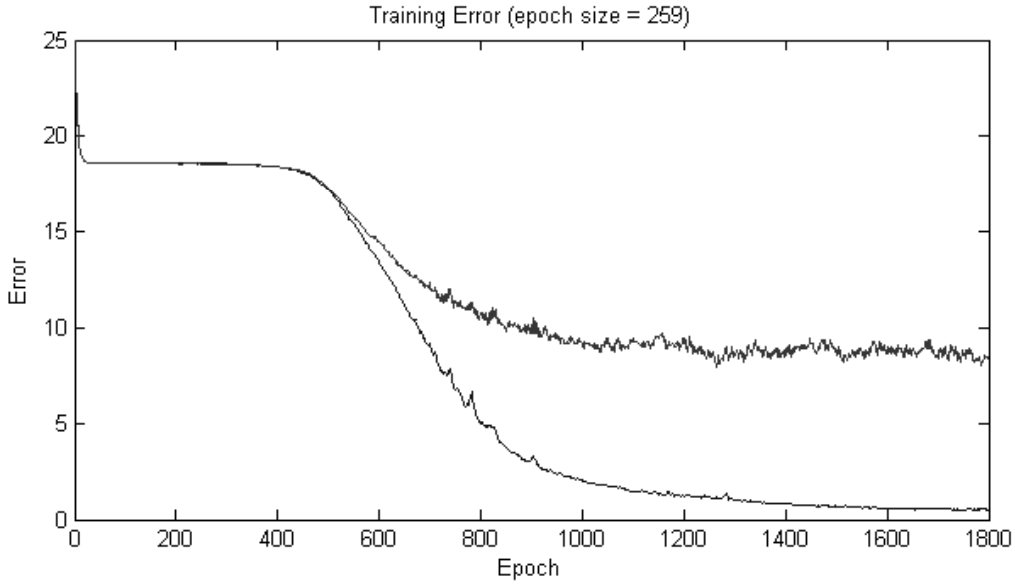


Figure 4.22 MSE of training and validation data sets with Early Stopped Method.

Results show that early stopping of iteration indeed has effect on our success. 5.5 % of increase is reached. As we mentioned before, we have 30 times more weight synapses then the number of sample inputs. Thus, early stopping method greatly affects our generalization ability of ANN.

4.5.3 Simulation of Network for Classification of Healthy and Pathological Gait

Data pool of the sample pattern is modified such that dimension of target output is appointed 2 instead of 4. One should note that Right Hemiplegia, Left Hemiplegia and Diplegia samples are implied as pathological gait pattern. Classification of “Healthy” and “Pathological” is accomplished. Neural Network architecture remained the same except for the output layer neuron number. Since we have only two groups to classify, number of neurons at output layer allocated is 2.

Table 4.20 Simulation results of classification for Right and Left Hemiplegia.

	TRAINING		TEST	
Estimated / Real	Healthy	Pathological	Healthy	Patological
Healthy	55	0	23	4
Patological	0	204	1	100
Clasification Success in %	100		96.1	

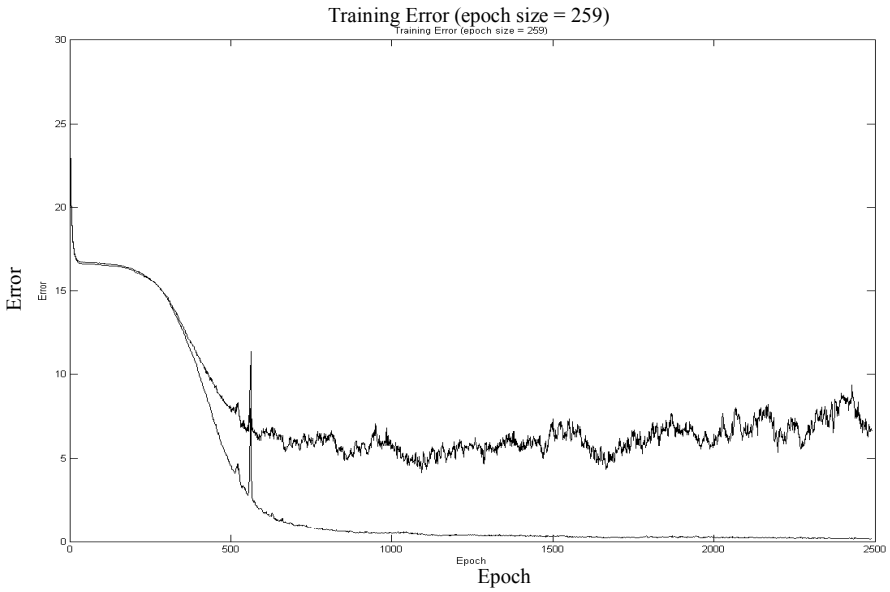


Figure 4.23 MSE of training and validation sets for 2 different gait types.

Our ANN performed really well to categorize distinct gait patterns. 96% of correct classification is achieved. This could be improved by means of Early stopping method.

5. DISCUSSION AND CONCLUSION

The aim of this project is to train Artificial Neural Network for classification of the four different gait patterns. Those are “Healthy”, Right Hemiplegia, Left Hemiplegia and Diplegia. Quest of the “best” learning constant and modifications done to architecture improved classification ability of it. Revision of data implemented shows that source impurity decreases network performance. Results of our study give rise to the future prospectus of ANN. There are still spaces to refinement of the network.

When carefully investigating average graphs of gait samples, one could be aware of asymmetric curves of the Left and Right sides. It is not prominent in both normal and diplegia but they are distinct in Right and Left Hemiplegia.

Early studies of simulation with ankle parameter show that it is impossible to categorize gait disorders after a while. We need more features that distinguish gait groups from each other. Five kinematical pair of data collected is used to improve classification success.

Network fed with more feature parameters performs better then with single parameter. When we compare four classifications done with only Ankle Dorsi-Plantar Flexion angle, there is remarkable increase in correct estimation. Although training attained 100% success, test outcomes were worse which was 47.7%. When we include all of the gait parameters, which we were collected for input matrix files, the results of correct classification are increased to 56.8%. Success is greater then previous one even though training was not 100% successful. Training results accomplished 67.8%, most probably algorithm stuck at either local minima or at plateau of the error surface (cost function). Another possible scenario is that Network design is not proper to perform further distinctions. Sharp convergence of the hidden layer’s neuron number from 1000 to 4 may lead to low classification. Discussion over that problem leads us to change network architecture at the hidden layer. Thus number of hidden layers increased from one to three. 1000 neurons for first layer, 80 neurons for second layer and 8 neurons for third layer were assigned. Results obtained were incredible 14% of increase

obtained in test results, plus complete convergence of training samples. Further architecture modifications are implemented but there were no significant improvements observed.

Neural architecture of the network kept as 1000 input node, 1000 neurons in first hidden layer, 80 neurons in second hidden layer, 8 neurons in third hidden layer and 4 neuron in output layer. Our performed experiments with different learning rate show that selected learning rate has optimum value, as it is stated in literature too. The smaller the learning rate parameter, the smaller the changes to the synaptic weights in the network will be from one iteration to the next, and the smoother will be the trajectory in weight space, however improvement is attained at cost of slower rate of learning. Additionally, making learning constant too large in order to speed up the rate of learning, results in large changes in the synaptic weights in such a form that the network may become unstable. Results of training of our ANN code approved that the best value of learning constant should be selected carefully for better performing Network. Network simulated for several learning constants, which are 0.2, 0.02, 0.002, 0.0002, and 0.00002. The best performance obtained when learning constant is equal to 0.002, and classification success was 70%.

Several trials of training of Network resulted in average 70% and no further improvements obtained. Thanks to my supervisors, we conclude that data selected has very wide range of ages, as it illustrated at the Table 3.1. Clinicians approved that great variations are observed in the gait patterns of various age groups. Thus specific age group is chosen for training of Network. Age group with greatest amount of gait data is chosen, which were between 6 and 12 years old. We had 150 subjects left out of 371 subjects after age confinement. Additionally, each subject's data were checked for video consistency with motion analysis data. Implementation of revised data may result in vast improvements in network training. Test of the network attained 78.4% of correct classification. In order to verify results, program executed 9 times with 9 different input file sets. Average of obtained results were 77.6 ± 1.7 success and range between 76.8-78.4.

Neural Network cross-validated by applying data set which was never seen before by network. The training session is stopped periodically (i.e., every so many epochs), and the network is tested on the validation subset after each period of training. More specifically, the periodic estimation-followed-by-validation process proceeds. It is observed that Mean Squared

Error curve of the validation data set decreased at the same time with training sample. At specific point error of the validation does not decrease, it even starts to increase. To keep generalization ability of NN training is terminated by means of early stopping method of

Table 5.1 Neural Network Classification Studies.

AUTHOR	NETWORK TYPE	CATEGORIES	BEST RESULT
Holzreiter and Köhle [20]	Feedforward (one hidden layer)	(1) Able-bodied gait (2) Pathological gait	Close to 95%
Barton and Lees [21]	Feedforward (two hidden layers)	(1) Normal walking (2) Leg length difference 20 mm thick sole (3) Exagerated lymphoedema 3.5 kg mass difference	83.3%
Lafuente et al. [18]	Feedforward (one hidden layer)	(1) healthy (2) ankle arthrosis (3) knee arthrosis (4) hip arthrosis	80%
This study	Feedforward (three hidden layers)	(1) Normal (2) Right Hemiplegia (3) Left Hemiplegia (4) Diplegia	85.2%

training. Training is terminated at 1800th epoch. 85.2 % of correct classification is obtained. We conclude that overtraining of our network decreases generalization ability. Early stopping of training increases NN performance in average by 4%.

Implemented ANN network show us how successful we were in our project. It is found out in literature that 80% of pattern classification is average. Those studies are shown in Table 5.1.

Holzreiter and Köhle achieved up to 95 % correctly classified samples. Which is great success when we compare it with other studies. They used coefficients of Fourier-transformation of Ground Reaction Force to classify Normal and Pathological pattern. In our study we used kinematic parameters and simulated designed network to classify two different gait conditions. As a result we achieved up to 96 % correctly classified samples that also show great consistency.

Barton and Lees work on three different gait patterns, which are simulation of clinical patient by means of attached 3.5 kg mass or 20 mm thick sole. In our study we used normal subjects and patients with real gait disorders: Right Hemiplegia, Left Hemiplegia and Diplegia. We achieved better results despite using 4 groups of subjects and real sample patterns without exaggeration.

Lafunte et al, on average 87 % of the control subjects were correctly classified, whereas only 73% the arthrosis patients were detected. Lafunte et al used kinetic data as opposed to kinematic data that was used in our study. Among these researches this one has the same number of classification groups as in our study. In spite of this fact in our research we achieved more accurate results.

Researches previously done proved that Artificial Neural Network is better classifier then statistical one [18]. It provides help to medical experts in decision support. Cost of the diagnosis is decreased by means of fast classification.

In this project we tried to categorize the walking conditions by the help of an artificial neural networks using kinematic data. Neural networks were trained to distinguish four gait patterns, which were normal walking, Right Hemiplegia, Left Hemiplegia and Diplegia.

Subsequent to training the NNs could recognize unknown gait patterns at a correct assignment percentage of 82.8%.

As number of classification groups increase success of ANN decreases. There were obtained results close to 95% success, however it was categorization only of Normal gait and Pathological gait groups. Implementation of the same problem to our NN concluded to 96% of correct categorization.

This study could help clinicians with different data classification and takes into account all details, which arise after motion analysis is performed. It is natural that a clinician considers only vital gait parameters and hardly evaluates all information due to insufficient time and resources. This neural classifier enables clinicians to base his/her judgment on all available data.

It could be beneficial to refine acquired data in order to raise the success of implementation. Aggregation is achieved through three normalization steps. Firstly mean of the data is removed. Afterwards obtained results are decorelated and finally covariance of the data is equalized.

Complexity of neural networks is directly related with the number of synaptic weights. As the input nodes increase (input dimension) corresponding weight number increases. This issue passes in literature as curse of dimensionality. Large quantity of sample patterns is indispensable for the convergence of synaptic weights of network. However it is hardly possible to attain so many data in the real world as required for training. Thus principal component analysis is deployed to reduce input dimension.

The results and studies made in literature encourage using ANN as decision support. Such a system could be linked to the kinematic analysis system, providing prompt evaluation of gait. As a result it is worth investigating for further increase of success of the classification.

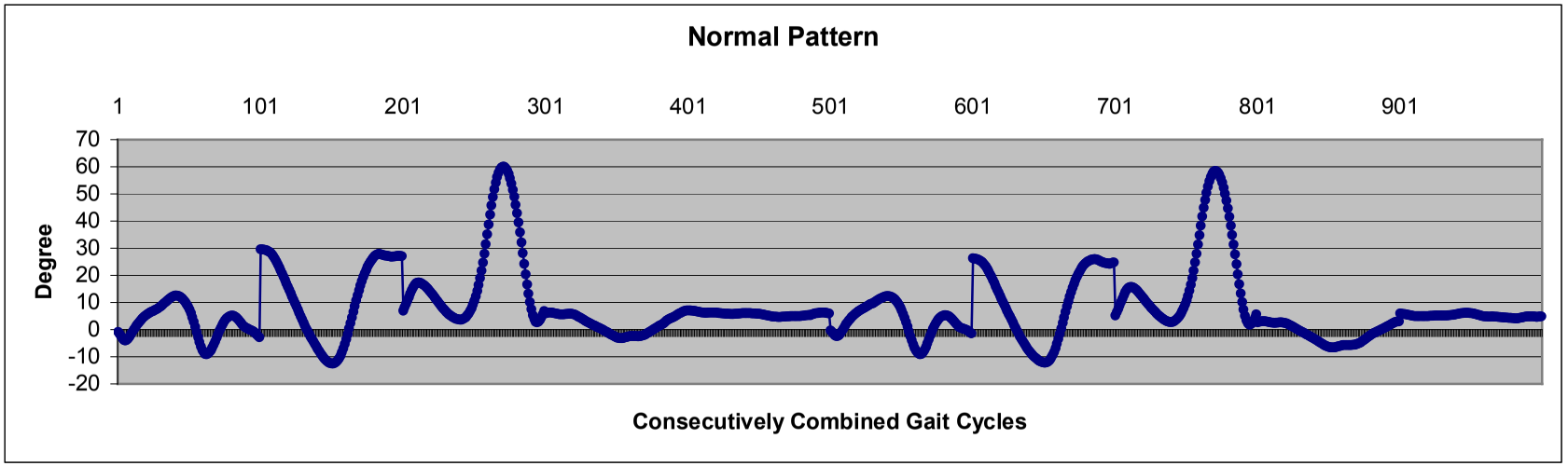
APPENDIX A:

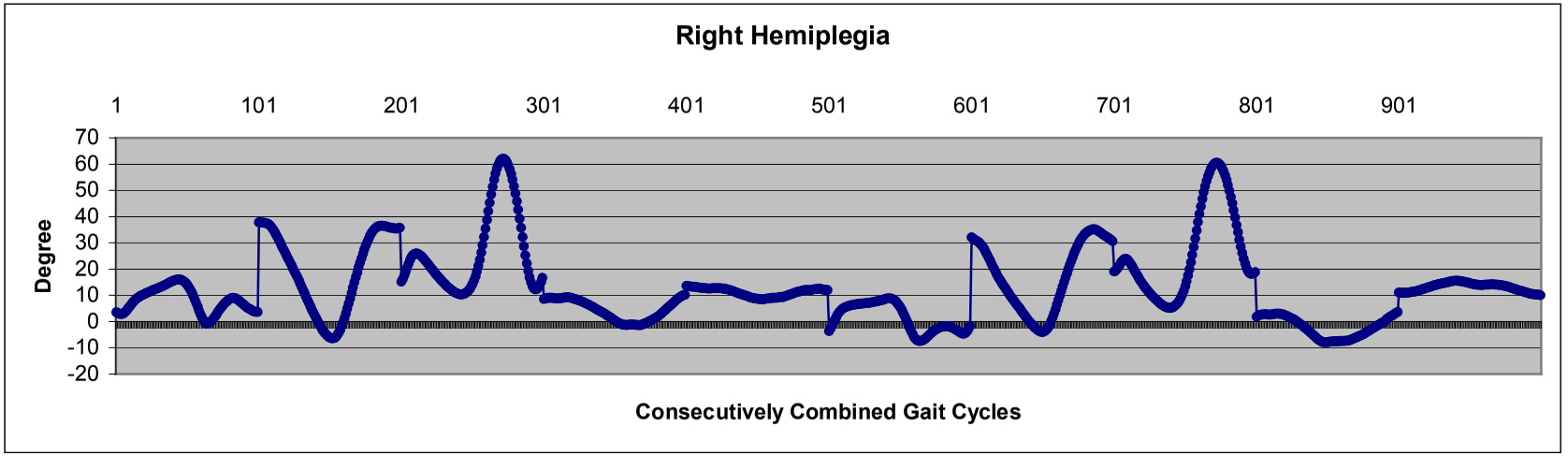
MARKERS POSITIONS OF THE LOWER EXTREMITY ACCORDING TO DAVIS MARKER PLACEMENT PROTOCOL

Abbr. Name	Full Name	Detailed Instructions
SACR	Sacrum	Keep the center of marker to be on the reversed direction of "i" vector of pelvis LCS
R, LASIS	Ant. Sup. Iliac spine	Keep the center of marker to be on the direction of "i" vector of pelvis LCS
R, LGTRO	Greater Trochanter	On the greater trochanter of femur
R, LTHI_W	Thigh wand	A stick marker on the lower lateral thigh. Should be on the plane that is formed by hip join, knee joint and lateral epicondyle marker.
R, LTHI_L	Thigh lateral	On the lateral thigh. The plane formed by RTHI_A, hip joint center and knee joint center will be sagittal plane of thigh.
R, LTHI_A	Thigh anterior	On the anterior thigh. Should be on the plane that is formed by hip join, knee joint and lateral epicondyle marker.
R, LLCON	Lateral Epicondyle	On the center of lateral epicondyle of femur. You can find it more easily if the subject's knee flexed a little.
R, LMCON	Medial Epicondyle	Only for static trial. On the center of medial epicondyle of femur. Can be found more easily if the subject's knee flexed a little.
R, LTIB_W	Tibial wand	A stick marker on the upper lateral surface of lower leg. Around upper 1/3 point of fibular head to lateral malleolus, where is no tibial torsion component.
R, LTTUB	Tibial tuberosity	On the tibial tuberosity
R, LFH	Fibular Head	On the fibular head. 2/3cm inferior to fibular head would be better to avoid merging with lateral condyle marker
R, LLMAL	Lateral Malleolus	Center of lateral malleolus
R, LMMAL	Medial Malleolus	Center of medial malleolus
R, LMT	Head of 2nd Metatarsal	On the second MT head
R, LHEEL	Heel	On heel.
R, LANKLE	Ankle (estimated)	Estimated position of ankle joint
R, LKNEE	Knee (estimated)	Estimated position of knee joint
R, LHIP	Hip (estimated)	Estimated position of hip joint

APPENDIX B:

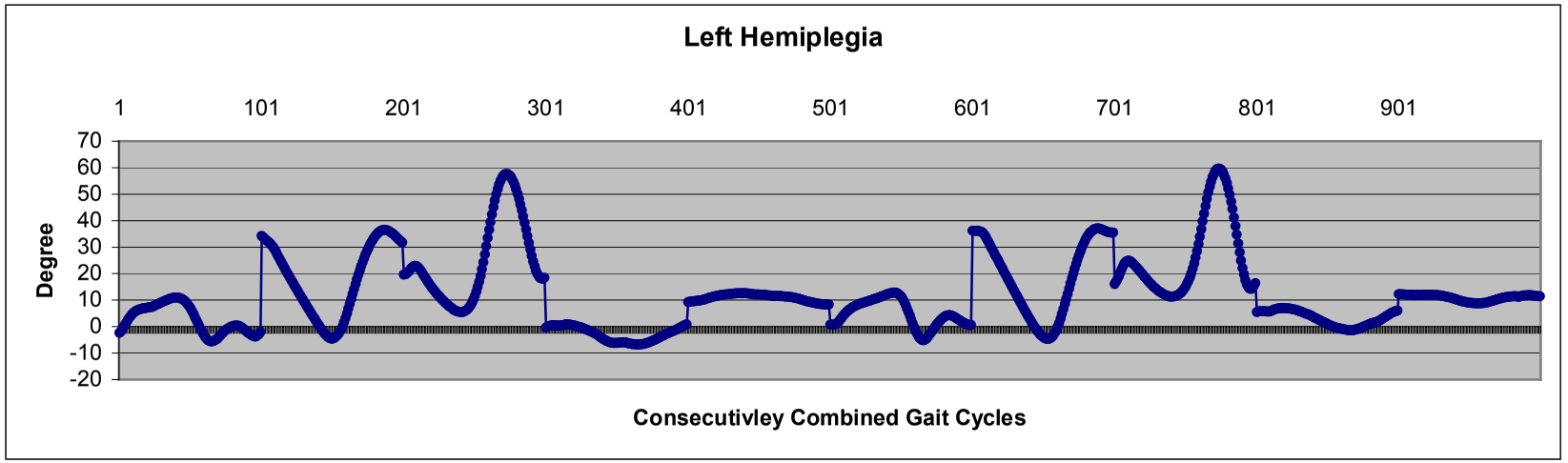
Graph Of Average Of Overall Data For Each Group

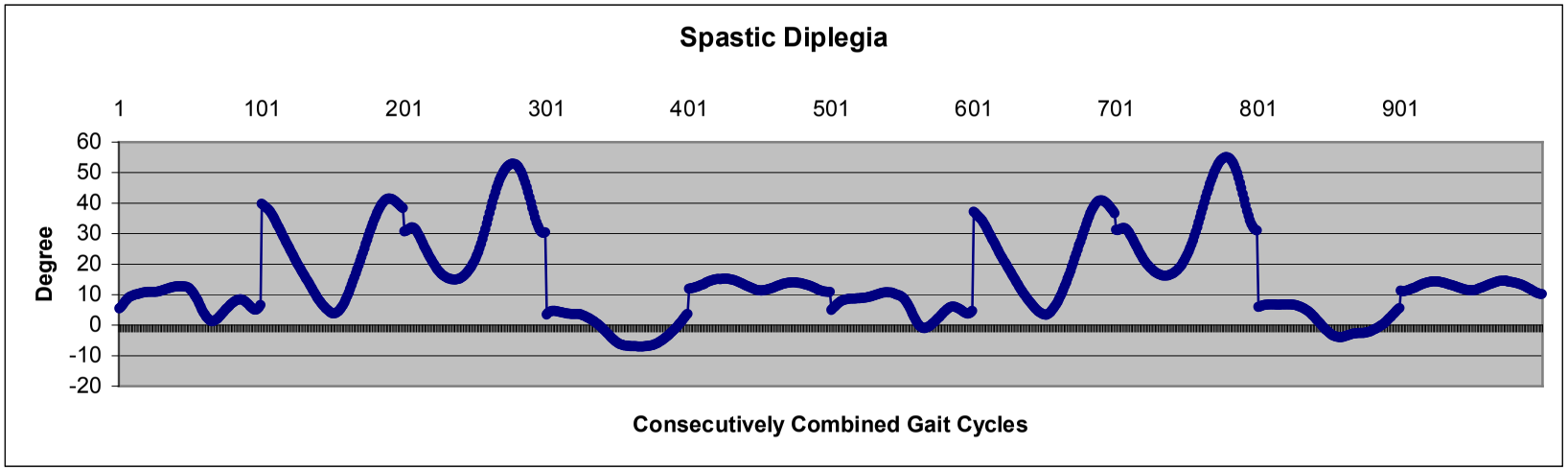




APPENDIX B: Continue

Graph Of Average Of Overall Data For Each Group





APPENDIX C:

RESULTS OF APPLIED NINE DIFFERENT DATA FILES TO NEURAL NETWORK

	TEST RESULTS # 1 CLASSIFICATION=76.8%				TEST RESULTS # 2 CLASSIFICATION=78.4%				TEST RESULTS # 3 CLASSIFICATION=78.4%			
	Estimated as Normal	Estimated as Diplegia	Estimated as R. Hemi	Estimated as L. Hemi	Estimated as Normal	Estimated as Diplegia	Estimated as R. Hemi	Estimated as L. Hemi	Estimated as Normal	Estimated as Diplegia	Estimated as R. Hemi	Estimated as L. Hemi
Real Samples as Normal	39	7	9	0	43	1	10	1	46	7	2	0
Real Samples as Diplegia	2	56	3	6	2	50	3	12	4	54	5	4
Real Samples as R. Hemi	8	5	34	10	7	2	38	10	13	2	39	3
Real Samples as Left Hemi	0	3	7	70	0	5	3	72	0	9	7	64
	TEST RESULTS # 4 CLASSIFICATION=79.6%				TEST RESULTS # 5 CLASSIFICATION=79.2%				TEST RESULTS # 6 CLASSIFICATION=78.4%			
Real Samples as Normal	47	4	4	0	47	5	3	0	46	6	3	0
Real Samples as Diplegia	6	49	2	10	0	58	0	9	5	50	2	10
Real Samples as R. Hemi	9	5	36	7	8	5	35	9	5	2	39	11
Real Samples as Left Hemi	0	2	4	74	0	7	8	65	0	2	10	68
	TEST RESULTS # 7 CLASSIFICATION=76.1%				TEST RESULTS # 8 CLASSIFICATION=74.1%				TEST RESULTS # 9 CLASSIFICATION=78%			
Real Samples as Normal	44	6	5	0	45	2	7	1	45	2	7	1
Real Samples as Diplegia	3	51	3	10	5	47	3	12	0	54	2	11
Real Samples as R.Hemi	7	6	31	13	5	2	37	13	5	5	34	13
Real Samples as Left Hemi	0	4	5	71	1	9	7	63	0	4	7	69

REFERENCES

1. Kirtley, C., Clinical Gait Analysis: Theory and Practice, Churchill Livingstone, 1st ed., 2006. Available: http://www.univie.ac.at/cga.
2. Motion Labs Systems, Inc., Baton Rouge, USA, 2005. Available: http://www.motion-labs.com/gait_analysis_benefits.htm.
3. Pathokinesiology Lab at Rancho Los Amigos Medical Center, Motion Lab Vicon System.
4. Motion Analysis Laboratory, Istanbul Medical Faculty, Istanbul University, Istanbul, Turkey, 2002.
5. Craig J. Newsam, Quantification of aquatic therapy water-based methods: Part II: Fine wire electromyography, The Journal of Aquatic Physical Therapy, Vol 4 pp 7-13, March 1996.
6. Gage, J. R., The Treatment of Gait Problems in Cerebral Palsy. London: Mac Keith Press, United Kingdom, 2004.
7. Winter, D. The biomechanics and motor control of human gait, Waterloo: University of Waterloo Press, 3rd ed., Canada, 2005.
8. Zatsiorsky, V. M., Kinematics of Human Motion, Champaign: Human Kinetics Publishers, 1st ed., USA, 1998.
9. Zatsiorsky, V. M., Kinetics of Human Motion, Champaign: Human Kinetics Publishers, 1st ed., USA, 2002.
10. Kirtley, C. Gait Analysis: Normal and Pathological Function, London: Slack Incorporated, 1st ed., Elsevier, UK, 1992.
11. Madan M. Gupta, Liang Jin, and Noriyasu Homma, Static and Dynamic Neural Networks From Fundamentals to Advanced Theory,New York: Addison-Wesley, USA, 2003
12. Haykin, Simon, Neural Networks: A comprehensive foundation, Prentice Hall, 2nd ed., USA, 1999.
13. Churchland, P.S., and T.J Sejnowski, The Computational Brain, Cambridge, MA: MIT Press 1992.
14. “Gait Abnormalities” , 2001. Available: http://www.orthoteers.co.uk

15. “Cerebral Palsy” , 2003. Available: http://health.allrefer.com/health/cerebral-palsy-info.html.

16. James A. Freeman, David M. Skapura Neural Networks Algorithms, Applications, and Programming Techniques, New York: Addison-Wesley, 1st ed., USA, 1991.

17. Minsky, M.L and S.A. Papert. Perceptrons, Cambridge: MIT Press 1969 Partially reprinted in nderson and Rosenfeld 1988.

18. R Lafuente, J M Belda, J Sanchez-Lacuesta, C Soler, J Prat, “Design and test of neural networks and statistical classifiers in computer-aided movement analysis: a case study on gait analysis”, Clinical Biomechanics, Vol. 13, pp. 216-229, 1997.

19. Hertz J, Krogh A, and Palmer R. G., Introduction to the Theory of Neural Computation, Santa Fe Institute, 1995.

20. Holzreiter SH., Köhle ME., “Assessment of gait patterns using neural network”,Journal of Biomechanics, Vol.26, pp.645-651, 1993.

21. Barton JG., Lees A., “An application of neural networks for distinguishing gait patterns on the basis of hip-knee joint angle diagrams”, Gait an Posture, Vol. 5, pp. 28-33, 1997.

22. L. Andrew Koman, Beth Paterson Smith, Jeffrey S.Shir. “Cerebral Palsy” The Lancet. Vol 363 May 15, 2004. pg 1619-1629.