

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ERKEN VENTRİKÜLER KASILMALARDA YSA
TABANLI BİR SINIFLANDIRICININ FPGA İLE
GERÇEKLEŞTİRİLMESİ**

**Tezi Hazırlayan
Ahmet Turan ÖZDEMİR**

**Tezi Yöneten
Prof. Dr. Kenan DANIŞMAN**

**Elektronik Mühendisliği Anabilim Dalı
Doktora Tezi**

**Ekim 2010
KAYSERİ**

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ERKEN VENTRİKÜLER KASILMALARDA YSA
TABANLI BİR SINIFLANDIRICININ FPGA İLE
GERÇEKLEŞTİRİLMESİ**

**Tezi Hazırlayan
Ahmet Turan ÖZDEMİR**

**Tezi Yöneten
Prof. Dr. Kenan DANIŞMAN**

**Elektronik Mühendisliği Anabilim Dalı
Doktora Tezi**

**Bu çalışma, Erciyes Üniversitesi Bilimsel Araştırma Projeleri Birimi
tarafından FBT-07-07 kodlu proje ile desteklenmiştir.**

**Ekim 2010
KAYSERİ**

Prof. Dr. Kenan DANIŞMAN danışmanlığında **Ahmet Turan ÖZDEMİR** tarafından hazırlanan “**Erken Ventriküler Kasılmalarda YSA Tabanlı Sınıflandırıcının FPGA İle Gerçekleştirilmesi**” adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü Elektronik Mühendisliği Ana Bilim Dalında **Doktora** tezi olarak kabul edilmiştir.

22/10/2010

(Tez Savunma Sınav Tarihi)

JÜRİ:

Başkan: Prof. Dr. Murat UZAM

Üye : Prof. Dr. Kenan DANIŞMAN

Üye : Prof. Dr. Mustafa ALÇI

Üye : Prof. Dr. Derviş KARABOĞA

Üye : Prof. Dr. Recai KILIÇ

ONAY:

Bu tezin kabulü Enstitü Yönetim Kurulunun 07/12/2010 tarih ve 2010/40-05 sayılı kararı ile onaylanmıştır.

07/12/2010


Prof. Dr. Necmettin MARAŞLI
Enstitü Müdürü

TEŞEKKÜR

Başta tez çalışmam boyunca bana yol gösteren, bilgi, tecrübe ve mesaisini benim için cömertçe harcayan danışman hocam sayın Prof. Dr. Kenan Danışman'a teşekkür ederim.

Tez izleme komitemde yer alan hocalarım sayın Prof. Dr. Mustafa Alçı ve sayın Prof. Dr. Derviş Karaboğa'ya çalışmalarına olan çok değerli katkılarından dolayı teşekkür ederim.

Beni birçok konuda yüreklendiren, hedeflerimi büyüten ve dünyaya daha geniş bir pencereden bakma becerisi kazandıran sayın Dr. H. Fatih Uğurdağ'a teşekkür ederim.

MIT Sea Grant College Programı kapsamında, burslu olarak MIT AUV laboratuvarında FPGA ile sistem tasarımı konusunda çalışmama imkan sağlayan ve eğitimlerime sponsor olan sayın Prof. Dr. Chrys Chrysostomidis'e ve bu süre içerisinde yardımcı danışmanlığımı yürüten sayın Dr. Milica Stojanovic'e teşekkür ederim.

Erciyes Üniversitesi'ndeki değerli arkadaşlarıma, gerek çalışmalarına gerekse yaşantıma olan pozitif katkılarından dolayı teşekkür ederim.

Desteklerinden ötürü Erciyes Üniversitesi Bilimsel Araştırma Projeleri Birimine teşekkür ederim.

Varlıkları benim en büyük mutluluğum olan sevgili anneme ve babama hayatımın her safhasında vermiş oldukları destekler için teşekkürü hayat boyu ödenemeyecek bir borç bilirim.

Sönmeyen ışığı ile bana ilham veren kardeşim ve en yakın arkadaşım Serdal Özdemir'e ve sevgili kız kardeşim Hatice Kuzucu'ya teşekkür ederim.

Özellikle de, aralıksız devam eden yoğun araştırma süreçlerinde beni destekleyen ve benden güler yüzünü hiçbir zaman eksik etmeyen sevgili eşim Ayşe Özdemir'e çok teşekkür ederim.

ERKEN VENTRİKÜLER KASILMALARDA YSA TABANLI BİR SINIFLANDIRICININ FPGA İLE GERÇEKLEŞTİRİLMESİ

Ahmet Turan ÖZDEMİR

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü

Doktora Tezi, Ekim 2010

Tez Danışmanı: Prof. Dr. Kenan DANIŞMAN

ÖZET

Kalp damar hastalıkları hem gelişmiş hem de gelişmemiş ülkelerde, erişkinlerde çeşitli organlarda fonksiyon bozukluklarına sebep olan temel etken ve en yaygın ölüm sebebidir. Dünya Sağlık Örgütü'nün (WHO, World Health Organization) verilerine göre her yıl 17,1 milyon insan kalp hastalıklarına bağlı nedenler sonucunda hayatını kaybetmektedir. Bu ölümlerin birincil sebebi, aritmi olarak isimlendirilen kalpteki elektriksel anormalliklerdir. Tehlikeli kalp bozukluklarının erken teşhisi, kalp hastalıklarının tedavisi ve ani kalp ölümlerinin önlenmesinde oldukça önemlidir. Bu bakımdan ElektroKardiyoGram (EKG) işaretlerinden aritmileri tanıyan otomatik aritmi sınıflandırıcıları erken teşhis adına hayati öneme sahiptir.

Son yirmi yıldır EKG işaretleri içerisindeki aritmileri tespit eden teknikler üzerine birçok çalışma yapılmıştır. Yüksek tahmin yeteneğinden dolayı Yapay Sinir Ağları (YSA), Medikal Tanı Sistemlerinde (MTS) en çok kullanılan yöntem olmuştur. Biyolojik sinir ağlarından esinlenerek geliştirilen YSA modeli, paralel mimarisi sayesinde hesaplamalarda oldukça hızlıdır. Fakat literatürde önerilen YSA tabanlı aritmi sınıflandırıcı modelleri, büyük boyutları sebebi ile ancak bilgisayarlarda, güçlü işlemciler üzerinde gerçekleştirilebilmektedir. Gerçek YSA modellerinin paralel donanımlar üzerinde gerçeklemesini mümkün kılmak için ağırlı girişine uygulanan EKG işaretinin öznitelik sayısı azaltılarak YSA boyutunun küçültülmesi gerekmektedir.

Bu tez çalışmasında, mevcut pek çok yazılım tabanlı YSA modeline karşın, taşınabilir bir YSA tabanlı otomatik aritmi sınıflandırıcının tek bir Alan Programlanabilir Kapı

Dizileri (FPGA, Field Programmable Gate Arrays) yongası üzerinde donanımsal olarak gerçekleştirilmesine odaklanılmıştır. Öznitelik çıkarım (boyut azaltma) yöntemi olarak Temel Bileşen Analizi (TBA) kullanılmış, YSA'nın boyutları küçültülmüş ve 8x2x1 boyutlu bir YSA tabanlı aritmi sınıflandırıcı, IEEE 754 32 bit kayan noktalı ve 16 bit sabit noktalı nümerik tanımlar kullanılarak sırası ile %97.66 ve %96.54 sınıflandırma başarımlarıyla tek bir FPGA yongası üzerinde oluşturulmuştur.

Anahtar Sözcükler: EKG, Aritmi, FPGA, YSA, TBA, Sınıflandırma.

AN FPGA IMPLEMENTATION OF ANN-BASED PREMATURE VENTRICULAR CONTRACTION CLASSIFIER

Ahmet Turan ÖZDEMİR

Erciyes University, Graduate School of Natural and Applied Sciences

Ph. D. Thesis, October 2010

Thesis Supervisor: Prof. Dr. Kenan DANIŞMAN

ABSTRACT

Cardio Vascular Disease (CVD) is one of the major causes of disability in adults as well as the main causes of death in both developed and undeveloped countries, and claiming 17.1 million lives a year according to the World Health Organization (WHO). CVD is the consequence of cardiac arrest and primarily caused by electrical abnormalities of the heart called arrhythmias. Early diagnosis of dangerous heart conditions is very important in treatment heart diseases and preventing sudden cardiac death. Therefore automatic ElectroCardioGram (ECG) arrhythmia classifiers are essential to timely diagnosis.

Many studies on recognition of cardiac arrhythmias from ECG using various techniques have been made in the past 20 years. Artificial Neural Network (ANN) is the most widely used tool in Medical Diagnosis Systems (MDS) because of their powerful prediction characteristics. An ANN model is inspired by real biological neural networks, with parallel structure that is potentially fast for computation. However, the suggested ANN architectures in the literature can only be run sequentially, on powerful processors, due to their complex architectures. To enable the implementation of real ANN models on parallel devices, the features of the ECG signal which are applied to the ANN inputs must be reduced in order to reduce ANN size.

In this thesis, it is focused on hardware implementation of a mobile ANN based automatic arrhythmia classifier on a single chip Field Programmable Gate Arrays (FPGA) instead of various software models of ANN. Using Principal Component

Analysis (PCA) as a feature extractor (size reductor) technique, ANN dimensions are reduced and 8x2x1 sized ANN-based arrhythmia classifier is implemented on a single chip FPGA, both IEEE 754 32 bit floating and 16 bit fixed point numerical representation with 97.66% and 96.54% classification accuracies, respectively.

Keywords: ECG, Arrhythmia, FPGA, ANN, PCA, Classification.

İÇİNDEKİLER

KABUL VE ONAY	i
TEŞEKKÜR.....	ii
ÖZET	iii
ABSTRACT	v
TABLolar LİSTESİ	xv
ŞEKİLLER LİSTESİ	xvi
1. BÖLÜM.....	1
GİRİŞ	1
1.1. Literatür Özeti	3
1.2. Çalışmanın Amacı.....	6
2. BÖLÜM.....	10
KALP VE ELEKTROKARDİYOGRAM	10
2.1. Kalbin Görevi ve Yapısı.....	10
2.2. Kalbin Elektriksel İletim Sistemi	11
2.3. Elektrokardiyogram (EKG).....	13
2.3.1. EKG'nin Tarihçesi [43].....	14
2.3.2. EKG'nin Elektriksel Özellikleri ve Ölçümü	14
2.4. Kalpteki Ritim Bozuklukları	15
2.4.1. Normal Sinüs Ritmi	16
2.4.2. Erken Karıncık Kasılması (PVC, Premature Ventricular Contraction) ..	16
2.4.3. Füzyon (Fusion) Aritmisi	18
2.5. MIT-BIH Aritmi Veritabanı.....	19
2.6. PhysioNet	20
2.7. Veri Okuma Yöntemi	21

3. BÖLÜM.....	22
EKG İŞARETLERİNİ SINIFLANDIRMADA KULLANILAN ÖZNİTELİK	
ÇIKARIM ALGORİTMALARI	22
3.1. Kalp Atımı Aralıkları ve Morfolojik Özellikler	24
3.2. Frekans Özellikleri	25
3.3. Dalgacık Dönüşümü (DD)	26
3.4. Temel Bileşen Analizi (TBA)	27
3.4.1. Tez Çalışmasında Kullanılan Kayıtların Özellikleri	28
3.4.2. TBA Yönteminin EKG Kayıtlarına Uygulanması	30
4. BÖLÜM.....	35
SINIFLANDIRMA YÖNTEMİ OLARAK YSA	35
4.1. YSA'nın Kullanım Alanları ve Tarihsel Gelişimi	36
4.2. Biyolojik Sinir Ağı	37
4.3. Yapay Sinir Ağlarının Özellikleri	40
4.3.1. Yapay Nöron Modeli	41
4.3.1.1. Aktivasyon Fonksiyonları	42
4.3.1.1.1. Logaritmik Sigmoid Aktivasyon Fonksiyonu	43
4.3.1.1.2. Doğrusal Aktivasyon Fonksiyonu	43
4.4. Biyolojik ve Yapay Sinir Ağlarının Kabiliyetleri	44
4.4.1. YSA'ların Sınıflandırılması	45
4.4.1.1. YSA'ların Ağ Yapılarına Göre Sınıflandırılması	45
4.4.1.1.1. İleri Beslemeli Yapay Sinir Ağları	46
4.4.1.1.1.1. Çok Katmanlı Perseptron	46
4.4.1.1.2. Geri Beslemeli Yapay Sinir Ağları	48
4.4.1.2. YSA'ların Öğrenme Yöntemlerine Göre Sınıflandırılması	49
4.4.1.2.1. Danışmanlı Öğrenme Metodu	49
4.4.1.2.2. Danışmansız Öğrenme Metodu	49
4.4.1.2.3. Takviyeli Öğrenme Metodu	49
4.4.2. YSA Öğrenme Algoritmaları	50
4.4.2.1. Geri Yayılım (BP) Algoritması	50
4.4.2.2. LM Algoritması	51

5. BÖLÜM.....	53
VERİ SETİNİN HAZIRLANMASI VE YSA MODELLERİNİN OLUŞTURULMASI.....	53
5.1. Veri Setinin Hazırlanması.....	54
5.2. Veri Setine TBA Uygulanması	62
5.3. MLP Tipi Yapay Sinir Ağının Eğitilmesi	64
5.4. 8x2x1 Boyutlu YSA'nın Ağırlıklarının Elde Edilmesi.....	74
6. BÖLÜM.....	77
PVC TİPİ ARİTMİ SINIFLANDIRICININ FPGA DONANIMININ OLUŞTURULMASI.....	77
6.1. FPGA ve Donanım Geliştirme Yazılımları.....	85
6.1.1. FPGA ile Tasarımın Avantajları	90
6.1.2. Kullanım Alanları.....	90
6.2. YSA'nın Sayısal Sistem Modellemesi	91
6.3. Nümerik Tanımlama	96
6.3.1. Sabit Noktalı Nümerik Tanımlama	98
6.3.2. Kayan Noktalı Nümerik Tanımlama.....	99
6.3.3. Sabit ve Kayan Noktalı Nümerik Tanımlamaların Karşılaştırılması	101
6.4. 8x2x1 Boyutlu YSA'nın FPGA Üzerinde Gerçeklenmesi.....	103
6.4.1. IEEE 754 32 Bit Kayan Noktalı Nümerik Tanımlama ile FPGA Üzerinde YSA Donanımı Gerçekleştirilmesi (FPANN).....	108
6.4.2. 16 Bit Sabit Noktalı Nümerik Tanımlama ile FPGA Üzerinde YSA Donanımı Gerçekleştirilmesi (FXANN).....	118
6.5. FPANN ve FXANN Donanımlarının Kaynak Kullanımları Bakımından Karşılaştırılması	125
7. BÖLÜM.....	128
SONUÇ VE TARTIŞMA	128
KAYNAKLAR	134

EKLER.....	147
A. 544 Değişik Boyutlu Ağın Eğitiminde Kullanılan Matlab Kodları	147
B. Parçalı Doğrusal Sigmoid Aktivasyon Fonksiyonunun Matlab Kodları.....	152
ÖZGEÇMİŞ.....	155

KISALTMALAR

Acc	Doğruluk (Accuracy)
Adaline	Adaptif Doğrusal Nöron
ADC	Analog Sayısal Dönüştürücü (Analog Digital Converter)
AE	Atriyal Kaçak Atımı
AFD	Ayrık Fourier Dönüşümü
AP	Erken Kulakçık Kasılması
ART	Adaptif Rezonans Teorisi
ASIC	Uygulamaya Özel Entegre Devre (Application Specific Integrated Circuit)
ANN	Yapay Sinir Ağı (Artificial Neural Network)
AV	Atrioventriküler Düğüm (Atrioventricular node)
BCD	İkili Kodlanmış Onlu (Binary Coded Decimal)
BP	Geri Yayılım Algoritması (Back Propagation)
BPM	Momentumlu Geri Yayılım (Back Propagation with Momentum)
bpm/sn	Saniyedeki Kalp Atımı Sayısı
BPR	Esnek Geri Yayılım (Resilient Back Propagation)
BR	Bayes Düzleştirme (Bayesian Regularization)
BYSA	Bulanık YSA (Fuzzy ANN)
CG	Conjugate Gradyent
CGF	Fletcher-Powell Conjugate Gradyent
CGP	Polak-Ribiere Conjugate Gradyent
CPS	Saniyede Hesaplanan Bağlantı (Connections Per Second)
CPU	Merkezi İşlem Birimi (Central Processing Unit)
DBD	Delta Bar Delta
DD	Dalgacık Dönüşümü (Wavelet Transform)
DDFS	Direkt Dijital Frekans Sentezleyici
DSP	Sayısal İşaret İşlemcisi (Digital Signal Processor)
EDBD	Genişletilmiş DBD (Extended DBD)
EGBP	Eşleştirmeli Gradyent Azaltma Geri Yayılımı
EKG	ElektroKardiyoGram
ESN	Echo State Networks

F	Füzyon Kalp Atımı (Fusion Hearth Beat)
F	Küsurat (Fraction)
FD	Fourier Dönüşümü (Fourier Transform)
FP	Kayan Noktalı Nümerik Tanımlama (Floating Point Numerical Definition)
FPAA	Alan Programlanabilir Analog Diziler (Field Programmable Analog Arrays)
FPANN	IEEE 754 32 bit Kayan Noktalı Nümerik Tanımlamanın Kullanıldığı YSA
FPGA	Alan Programlanabilir Kapı Dizileri (Field Programmable Gate Arrays)
fPN	Paced ve Normal Atım Füzyonu (fusion of Paced and Normal beats)
FW	İleri Beslemeli (FeedforWard)
FX	Sabit Noktalı Nümerik Tanımlama (FiXed Point Numerical Definiton)
FXANN	16 bit sabit Noktalı Nümerik Tanımlamanın Kullanıldığı YSA
GA	Genetik Algoritma
GNA	Gauss Newton Algoritması
HDL	Donanım Tanımlama Dili (Hardware Description Language)
HMM	Hidden Markov Model
I	Tamsayı (Integer)
LBBB	Sol Dal Bloğu Atımı
LSB	En Düşük Değerlikli Bit (Least Significant Bit)
LM	Levenberg Marquardt Algoritması
Logsig	Logaritmik Sigmoid Aktivasyon Fonksiyonu
LPC	Doğrusal Kestirim Katsayısı (Linear Prediction Coefficient)
LUT	Look Up Table Yöntemi
LVQ	Linear Vector Quantization
Madaline	Multiple Adaline
MDS	Medical Diagnosis System
MIT-BIH	Massachusetts Institute of Technology-Belt Israel Hospital
ML II	Modified Lead II
MLP	Çok Katmanlı Perseptron (Multi Layer Perceptron)
MSB	En Yüksek Değerlikli Bit (Most Significant Bit)
MTS	Medikal Tanı Sistemi

N	Normal Kalp Atımı
NAF	Doğrusal Olmayan Aktivasyon Fonksiyonu (Nonlinear Activation Function)
NAND	Ve Değil Kapısı
NE	Nodal Kaçak Atımı (Nodal Escape Beats)
nE	Toplam Yanlış Sınıflandırılmış Kalp Atım Sayısı
NOR	Veya Değil Kapısı
NP	Nodal Erken Kasılma (Nodal Premature Contraction)
nT	Toplam Kalp Atım Sayısı
P	Paced Atım
<i>P</i>	Nümerik Hassasiyet
PCB	Baskılı Devre Kartı (Printed Circuit Board)
Purelin	Doğrusal Aktivasyon Fonksiyonu (Matlab Komut İsmi)
PVC	Erken Karıncık Kasılması (Premature Ventricular Contraction)
PWLAN	Parçalı Doğrusal Aktivasyon Fonksiyonu (Piece-Wise Linear Activation function)
Q	Tanımlayamayan Kalp Atımı
QRS	Q-R-S İntervalleri Bileşke Vurusu
Radbas	Radyal Aktivasyon Fonksiyonu
RBBB	Sağ Dal Bloğu Atımı (Right Bundle Branch Block)
RBF	Radyal Tabanlı Fonksiyon (Radial Basis Function)
RR	R Dalga Tepeleri Arasındaki Uzaklık
SA	SinoAtriyal Düğüm
Satlin	Doğrusal Saturasyon Aktivasyon Fonksiyonu
SDD	Sürekli Dalgacık Dönüşümü (Continuous Wavelet Transform)
SoC	System on Chip
SOM	Self Organization Maps
SP	Supraventriküler Kasılma
SVM	Destek Vektör Makinesi (Support Vector Machine)
Tansig	Tanjant Sigmoid Aktivasyon Fonksiyonu
TB	Temel Bileşen
TBA	Temel Bileşen Analizi (Principal Component Analysis)
Tribas	Üçgen Aktivasyon Fonksiyonu

U	Sınıflandırılmamış Kalp Atımı
XOR	Özel Veya Problemi
V	Ektopik Karıncık Kasılması (Ectopic Ventricular Contraction)
VE	Ventriküler Kaçak Atım (Ventricular Escape Beat)
VF	Ventriküler Fibrilasyon (Ventricular Fibrillation)
VT	Ventriküler Taşikardi (Ventricular Tachycardia)
VGA	Video Grafik Dizisi (Video Graphics Array)
VHDL	Çok Hızlı Tümleşik Devre Donanım Tanımlama Dili (Very High Speed Integrated Circuit Hardware Description Language)
VLSI	Çok Büyük Boyutlu Entegre Devre (Very Large Scale Integration)
VT	Ventriküler Taşikardi (Ventricular Tachycardia)
WFDB	Veritabanı Görüntüleme (WaveForm DataBase)
WHO	Dünya Sağlık Örgütü (World Health Organization)
YSA	Yapay Sinir Ağı (Artificial Neural Networks)

TABLOLAR LİSTESİ

Tablo 2.1. AAMI atım sınıfları ve MIT-BIH içeriği [5]	19
Tablo 3.1. Sağlıklı erişkin bir erkekte EKG özellikleri [42].....	23
Tablo 3.2. Tez çalışmasında kullanılan ham dosyaların aritmi içerikleri.....	29
Tablo 4.1. Biyolojik nöron ile yapay nöron arasındaki ilişki.....	42
Tablo 4.2. Beyin ile bilgisayar arasındaki performans ilişkisi [79]	45
Tablo 5.1. Eğitim veri setlerinin oluşturulması.....	58
Tablo 5.2. CGB algoritması ile eğitilmiş 544 adet değişik ağın test hataları.....	66
Tablo 5.3. CGF algoritması ile eğitilmiş 544 adet değişik ağın test hataları	67
Tablo 5.4. CGP algoritması ile eğitilmiş 544 adet değişik ağın test hataları	68
Tablo 5.5. SCG algoritması ile eğitilmiş 544 adet değişik ağın test hataları	69
Tablo 5.6. BPR algoritması ile eğitilmiş 544 adet değişik ağın test hataları	70
Tablo 5.7. LM algoritması ile eğitilmiş 544 adet değişik ağın test hataları.....	71
Tablo 5.8. 8x2x1 boyutlu YSA için ağ parametreleri FPANN	76
Tablo 6.1. Marketteki nöro donanımlar [140].....	94
Tablo 6.2. Mantıksal XOR problemini çözmekte kullanılan yapılar ve sonuçlar	102
Tablo 6.3. Cyclone III ailesinden bazı FPGA'ların donanım kaynak özellikleri.....	104
Tablo 6.4. FPANN ve ana bloklarının donanım kaynak kullanımları	111
Tablo 6.5. FPANN içerisindeki bazı önemli birimlerin kaynak kullanımları.....	111
Tablo 6.6. 8x2x1 boyutlu YSA için ağ parametreleri FPANN	118
Tablo 6.7. İlk sekiz temel bileşen	118
Tablo 6.8. FXANN içerisindeki bazı önemli birimlerin kaynak kullanımları	122
Tablo 6.9. FXANN ve ana bloklarının donanım kaynak kullanımları.....	124
Tablo 6.10. FPANN ve FXANN donanımlarının ve bu donanımlardaki temel blokların kaynak kullanımları bakımından karşılaştırılması	125
Tablo 6.11. FPANN ve FXANN donanımlarındaki aritmetik birimlerin kaynak kullanımları bakımından karşılaştırılması	126

ŞEKİLLER LİSTESİ

Şekil 1.1. Normal bir QRS bileşke vurusunda önemli noktalar ve aralıklar.....	2
Şekil 2.1. Kalbin iç yapısı ve önemli damarlar [39]	11
Şekil 2.2. Kalbin iletim sistemi ve uyarının yayılımı	13
Şekil 2.3. Elektrotların deri üzerindeki yerleşimi [42].....	15
Şekil 2.4. Normal sinüs ritmi	16
Şekil 2.5. Erken karıncık kasılmasına sebep olan odak kayması.....	17
Şekil 2.6. Erken karıncık kasılması [44]	18
Şekil 2.7. Füzyon (F) atımı	18
Şekil 2.8. WAVE programı ekran görüntüsü	21
Şekil 3.1. EKG işaret işlemede izlenen adımlar.....	22
Şekil 3.2. EKG işaretleri ile ilgili bazı tanımlamalar	23
Şekil 3.3. EKG işaret işlemede izlenen adımlar.....	29
Şekil 4.1. Biyolojik nöron [99]	39
Şekil 4.2. Merkezi sinir sisteminde veri akışı	40
Şekil 4.3. İşlemci eleman	41
Şekil 4.4. Logaritmik sigmoid aktivasyon fonksiyonu	43
Şekil 4.5. Purelin aktivasyon fonksiyonu.....	43
Şekil 4.6. İleri beslemeli YSA	46
Şekil 4.7. MLP ağ yapısı.....	47
Şekil 4.8. Geri beslemeli YSA	48
Şekil 5.1. 205, 208, 210 ve 215 numaraları kayıtlar içerisindeki N sınıfından veriler ...	54
Şekil 5.2. 205, 208, 210 ve 215 numaraları kayıtlar içerisindeki V sınıfından veriler ...	55
Şekil 5.3. 205, 208, 210 ve 215 numaraları kayıtlar içerisindeki F sınıfından veriler....	55
Şekil 5.4. İkincil veri seti içerisindeki N sınıfından veriler	56
Şekil 5.5. İkincil veri seti içerisindeki F sınıfından veriler.....	57
Şekil 5.6. İkincil veri seti içerisindeki V sınıfından veriler	57
Şekil 5.7. Nihai veri seti.....	59
Şekil 5.8. Normalize veri seti.....	61
Şekil 5.9. YSA tabanlı PVC sınıflandırıcının tasarım akış şeması	62

Şekil 5.10. İlk yirmi temel bileşen	63
Şekil 5.11. MLP YSA ağ değişkenliği	64
Şekil 5.12. İlk sekiz temel bileşen.....	73
Şekil 5.13. 8x2x1 YSA modeli	75
Şekil 6.1. Temel PAL mimarisi	79
Şekil 6.2. GAL22V10 içerisindeki programlanabilir Macrocell.....	80
Şekil 6.3. Temel CPLD mimarisi	81
Şekil 6.4. İç bağlantılar	82
Şekil 6.5. FPGA iç bağlantıları	82
Şekil 6.6. LE içyapısı	83
Şekil 6.7. FPGA ile tasarım süreci	86
Şekil 6.8. Altera DSP Builder ile Simulink ortamında tasarlanmış bir model.....	87
Şekil 6.9. Dört bit paralel çarpıcı devresi.....	88
Şekil 6.10. Dört bit paralel çarpıcının şematik sembolü	89
Şekil 6.11. 16 bit sabit noktalı nümerik sayı formatı	98
Şekil 6.12. 32 bit kayan noktalı nümerik sayı formatı	100
Şekil 6.13. EKG kayıtlarının FPGA modeline uygulanması	106
Şekil 6.14. Sekiz girişli bir işlemci elemanın blok diyagramı	106
Şekil 6.15. a) FPANN b) FXANN modelleri.....	107
Şekil 6.16. FPANN ana tasarım dosyası	109
Şekil 6.17. Kaynak kullanımları ve tasarım süreçlerinin onayı (FPANN)	110
Şekil 6.18. MemInit bloğunun içyapısı.....	112
Şekil 6.19. Generator bloğu tarafından üretilen ENABLE kontrol işareti.....	112
Şekil 6.20. 8x1N bloğunun içyapısı	113
Şekil 6.21. FPANN donanımının fonksiyonel benzetimi.....	114
Şekil 6.22. FPANN Simulink modeli.....	115
Şekil 6.23. Sigmoid aktivasyon fonksiyonu bloğu blok diyagramı	116
Şekil 6.24. Sigmoid aktivasyon fonksiyonu bloğunun içyapısı	116
Şekil 6.25. 2x1N bloğunun içyapısı	117
Şekil 6.26. PWLAN bloğunun içyapısı.....	120
Şekil 6.27. PWLAN fonksiyonunun gerçek sigmoide yakınsaması	121
Şekil 6.28. PWLAN fonksiyonunu ile gerçek sigmoid arasındaki fark.....	122

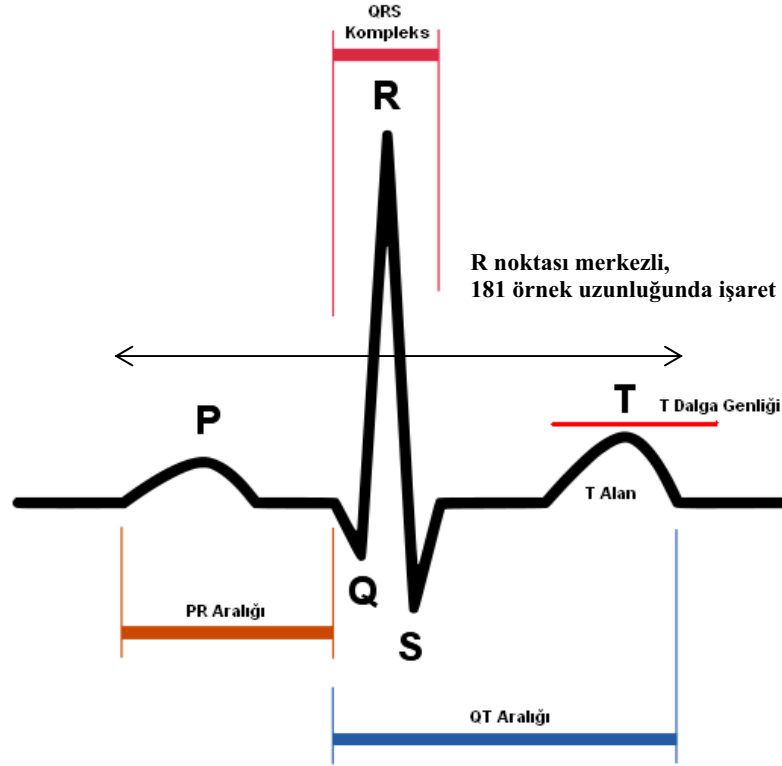
Şekil 6.29. Kaynak kullanımları ve tasarım süreçlerinin onayı (FXANN).....	123
Şekil 6.30. a) FPANN Chip Planner çıktısı, b) FXANN Chip Planner çıktısı.....	127

1. BÖLÜM

GİRİŞ

Kalp hastalıkları, erişkinlerde çeşitli organlarda fonksiyon bozukluklarına sebep olan temel etken ve en yaygın ölüm sebebidir. Dünya Sağlık Örgütü'nün (WHO, World Health Organization) verilerine göre, Avrupa'da her yıl yaklaşık iki milyon [1] dünyada ise 17,1 milyon insan kalp hastalıklarına bağlı nedenler sonucunda hayatını kaybetmektedir [2]. Bu ölümlerin birincil sebebi, aritmi olarak isimlendirilen kalpteki elektriksel anormalliklerdir [3]. Tehlikeli kalp bozukluklarının erken teşhisi, kalp hastalıklarının tedavisi ve ani kalp ölümlerinin önlenmesinde oldukça önemlidir. Erken teşhis, tedavi süreçlerine ve yöntemlerine de olumlu katkılarda bulunmakta ve hastaların serviste kalma sürelerini ve tedavi masraflarını da azaltmaktadır. Bu bakımdan ElektroKardiyoGram (EKG) işaretlerinden aritmileri tanıyan otomatik aritmi sınıflandırıcıları, erken teşhis adına hayati öneme sahiptir.

EKG insan kalbinin birim zaman içerisinde oluşturmuş olduğu biyoelektrik potansiyellerin bir toplamıdır ve kalp hastalıkların teşhis ve tedavisinde kullanılan, kalbin durumu hakkında bilgiler veren en önemli referanstır [4]. Vücut içerisine girilmeden deri üzerine belli bir sırayla yerleştirilen iletken plakalar üzerinden alınan potansiyel farkların ölçümü yoluyla elde edilen EKG, güvenilir ve hızlı bir tanı yöntemidir. EKG kayıtları Şekil 1.1'de gösterilen QRS bileşke vurusu ile yorumlanmaktadır. EKG işaretlerinin yazılım tabanlı sistemler ile yorumlanmaya başlanması 1960'lı yıllara rast gelir. EKG ile ilgili çalışmalar kalbin ritmik karakteristiği incelenerek yeni tanı ve tedavi sistemikleri geliştirmek için başlamış olup günümüzde klinik amaçlı olarak hekimlere yardımcı birçok yazılım ve cihaz kullanılmaktadır. Kalbin ritmik karakteristiğindeki bozuklukları yorumlayan yazılım ve cihazlar, EKG kayıtları içerisindeki aritmileri tespit ederek tanıya katkıda bulunurlar.



Şekil 1.1. Normal bir QRS bileşke vurusunda önemli noktalar ve aralıklar.

Kalpte aritmi meydana geldiği zaman kalp düzenli olarak kan pompalamaz. Sonuçta vücudu ve beyni yeterli miktarda kan ve oksijen ile besleyemez. Aritmiler iki gruba ayrılırlar, birinci grup aritmiler Ventriküler Taşikardi (VT, Ventricular Tachycardia) ve Ventriküler Fibrilasyon (VF, Ventricular Fibrillation) gibi ani ölümlere sebep olan ritim bozukluklarıdır. VT veya VF meydana geldiğinde, kişi saniyeler içerisinde şuurunu kaybeder ve ölüm dakikalar içerisinde gerçekleşir. O yüzden bu tür aritmiler çok kısa bir süre içerisinde defibratör cihazı ile sonlandırılmalıdır [5]. Literatürde VT ve VF tipi aritmileri etkili bir şekilde tanıyan otomatik sınıflandırıcılar hakkında çok sayıda çalışma bulunmaktadır [6-9]. İkinci grup aritmiler ise Erken Karıncık Kasılması (PVC, Premature Ventricular Contraction) gibi defibratör ile anında müdahaleyi gerektirmeyen fakat ileride oluşturacakları risklerden dolayı tedavi edilmesi gereken ritim bozukluklarıdır. Son yıllarda yapılan çalışmalar PVC tipi aritminin, artan kardiyak ani ölüm riskinin bir habercisi olduğunu göstermektedir [10]. Kalp hastalıklarına bağlı ölümlerin birçoğu, EKG kayıtları üzerinden aritmilerinin ön izlenmesi ve teşhisi ile fark edilip önlenmektedir [11].

PVC uzun holter kayıtlarından veya EKG çekildiği esnada tespit edilebilir. PVC düzensiz olarak görüldüğünden, işaretin analiz edilmesi ve PVC tespiti için bazen 24 saatlik holter kaydına ihtiyaç duyulabilir. Uzun EKG kayıtlarından aritmi çıkarımı çok zaman alabilir ve günlük iş yükünden kaynaklı yorgunluğa bağlı olarak, kardiyolog uzun EKG kaydındaki bazı önemli anormallikleri gözden kaçırabilir. Dahası kalp rahatsızlığı şikâyetiyle bir hastaneye veya bir sağlık kuruluşuna başvuran kişinin kardiyolog tarafından anında değerlendirilmesi oldukça zor bir ihtimaldir. Uzman hekim sayısı buna müsaade etmediği gibi istihdamları da hastaneler için oldukça maliyetlidir. Bu tür hastalar, hastanelere müracaat ettikleri ilk anda genellikle acil servis hekimleri tarafından verilen ön tanıyla işleme alınırlar. Uzun süreli EKG kayıtlarının değerlendirilmesinde, kardiyologun gün içindeki yoğunluğuna bağlı olarak meydana gelebilecek hataları indirmeye yardımcı olmak ayrıca kardiyolog bulunmayan sağlık kuruluşlarında ön tanıya fikir vermesi amaçları ile hekimlere yardımcı, otomatik aritmi tespit yazılımları kullanılmaktadır [12, 13].

1.1. Literatür Özeti

Otomatik PVC sınıflandırıcılar literatürde geniş anlamda çalışılmış bir konudur ve değişik teknikler kullanarak farklı başarımlarla sonuçlar üreten birçok teknik mevcuttur. Bunlar başlıca Yapay Sinir Ağları (YSA) [14-21], Genetik Algoritma (GA) [22], Gizli Markov Modelleri (HMM, Hidden Markov Models) [23], Doğrusal Ayırtaç Fonksiyonu (Linear Discriminant Function) [24], Adaptif Filtre [25], Destek Vektör Makinesi (SVM, Support Vector Machine) [26], Doğrusal Kestirim (Linear Prediction) [27], Bayes Tahmini (Bayesian Estimation) [10] ve istatistiksel [5] tabanlı teknikler olarak verilebilir. Bu teknikler EKG işaretlerinin birçok değişik özneliğini farklı yöntemler ile öznitelik setine (feature vector) çevirirler. Öznitelik çıkarma yöntemlerine örnek olarak Kalp-Atımı Aralıkları (Heart Beats Intervals) [5, 15, 18, 23, 24, 28], frekans özellikleri [25, 29], Dalgacık Dönüşümü (DD) [14, 15, 19, 30, 31], Hermit Çokluterimlileri (Hermite Polynominals) [18] ve Temel Bileşen Analizi (TBA, Principal Component Analysis) [16, 21, 32] yöntemleri örnek verilebilir. Güçlü tahminsel yeteneğinden dolayı Medikal Tanı Sistemlerinde (MTS) en çok kullanılan yöntem YSA'lardır. Birçok araştırmacı YSA'nın performansını diğer metotlar ile karşılaştırmış ve YSA'nın diğerlerinden daha iyi sonuçlar ürettiğini göstermişlerdir [33].

Literatürdeki PVC aritmi sınıflandırıcıları, birçok öznitelik çıkarım tekniği ile elde edilmiş öznitelik setlerinin, uygun bir sınıflandırma tekniğiyle işlenmesi ile değişik başarımlarda sınıflandırma performansları sergilemişlerdir. Sınıflandırma başarımlarının değerlendirilmesinde kullanılan en önemli metrik, Denklem 1.1’de verilen ağın doğruluğu parametresidir. Burada nT ve nE sırası ile bir kayıt içerisindeki toplam kalp atım sayısını ve toplam yanlış sınıflandırılmış kalp atım sayısını ifade etmektedir.

$$Acc = \frac{nT - nE}{nT} \times 100 \quad (1.1)$$

Hu ve arkadaşları [34] 51 örnek uzunluklu kalp atımlarını hiçbir öznitelik çıkarım yöntemi kullanmadan Çok Katmanlı Perseptron (MLP, Multi Layer Perceptron) YSA’ya giriş olarak uygulamışlardır. 13 değişik aritmi tipini sınıflandırmak amacı ile oluşturulan değişik boyutlardaki YSA’lardan 51x40x13 boyutlu bir ağ ile ortalama %90 doğrulukla sınıflandırma başarımı sağlamışlardır. Bu ağın PVC doğruluğu ise %75,6’dır.

Özbay ve arkadaşları [20] çalışmasında, Geri Yayılım (BP, Back Propagation) algoritması ile eğitilen bir MLP YSA ile 10 değişik tip aritmeyi sınıflandırmışlardır. MIT-BIH (Massachusetts Institute of Technology-Belt Israel Hospital) EKG veritabanından [35] kullandıkları kayıtlara herhangi bir öznitelik çıkarım yöntemi uygulamadan, R-R aralıklarını 200 örnek vektörü uzunluğunda örneklenmişlerdir. İçerisinde PVC tipi aritmi bulunmayan 10 tip aritmeyi 200x15x10 boyutlu bir YSA ile ortalama %97,8 doğrulukla sınıflandırmışlardır.

Bazı öznitelik çıkarım teknikleri, orijinal EKG işaretini yüksek bir yakınsama ile ifade edebilmek için çok fazla sayıda öznitelik setine ihtiyaç duyar. Kalp atım aralıkları, zaman, frekans ve genlik özellikleri bunlara birer örnektir. Fazla sayıdaki öznitelik YSA giriş katmanını büyütür ve ağın boyutunun artmasına sebep olur. Artan boyut ise hesaplama karmaşıklığına, ağdaki işlemci eleman sayısının ve hesaplama süresinin artmasına sebep olur. Dokur ve arkadaşları [36] DD ve Ayrık Fourier Dönüşümü (AFD) tekniklerinin performanslarını kıyaslarlarken, 15 DD katsayılı 11 noda sahip bir ağ ile %97, 15 AFD katsayılı 18 noda sahip bir başka ağ ile %89,4 doğrulukta sınıflandırma yapmışlardır. Sonuçlar, doğru bir öznitelik çıkarım metodu ile daha az sayıda öznitelikle yüksek doğrulukta sınıflandırma yapılabileceğini göstermektedir.

DD daha iyi sonuçlar vermesine karşın genellikle Bulanık (Fuzzy) Yapay Sinir Ağları (BYSA) tabanlı PVC sınıflandırıcılarında kullanılırlar. Bununla birlikte literatürdeki BYSA tabanlı PVC sınıflandırıcılar genellikle birçok noda ve üyelik fonksiyonuna sahip, en az üç veya dört katmanlı yapıdadırlar. Bu büyüklüklerdeki BYSA'ların FPGA üzerinde gerçekleştirilmesi için çok fazla sayıda donanım kaynağına ihtiyaç duyulur. Lim [14] öznitelik setlerinin BYSA tabanlı aritmi sınıflandırıcıların performanslarına olan etkilerini incelemiş ve BYSA boyutlarının azaltılması üzerine yoğunlaşmıştır. 8 DD katsayılı 3 katmanlı bir BYSA ile maksimum %99,8 doğrulukta PVC sınıflandırması yapmıştır. Bölgesel minimumdan kaçınmak amacı ile her bir üyelik fonksiyonunda, üç farklı bulanık küme kullanmış fakat ağdaki toplam nod sayısını vermemiştir. Ham ve arkadaşları [17] bir ortalama karesel değer ve iki Doğrusal Kestirimci Katsayısı (LPC, Linear Predictive Coefficients) olmak üzere toplam 3 öznitelik seti ile BYSA tabanlı bir aritmi sınıflandırıcıyı maksimum %96,995 doğruluk ile oluşturmuşlardır. Shyu ve arkadaşları [19] DD öznitelik çıkarım yöntemiyle, iki noda sahip bir giriş katmanı, altı noda sahip bir üyelik katmanı ve dokuz noda sahip bir kural katmanı olan bir BYSA ile %99,7 doğrulukla PVC sınıflandırması yapmışlardır.

İnan ve arkadaşları [15] daha iyi bir sınıflandırma başarımı elde edebilmek için EKG morfolojik özellikleri ile zaman özelliklerini birlikte kullanmışlardır. Bu çalışmada DD katsayıları ve R-R aralıklarının süreleri birleştirilerek bir öznitelik veri seti oluşturulmuştur. 22, 26, 32, 43 ve 64 örnek uzunluklu değişik öznitelik vektörleri kullanarak içerisinde PVC'nin de bulunduğu altı çeşit aritmiyi bir MLP YSA kullanarak sınıflandırmışlardır. En iyi sınıflandırma başarısı, 42 DD katsayısı ve bir R-R aralığı verisinden oluşan bir öznitelik seti ile en yüksek %95,2 oranında bir doğrulukla tespit edilmiştir. Bu doğruluk değeri 43 nöronlu tek bir gizli katmanı bulunan bir ağ ile sağlanmıştır.

Vargas ve arkadaşları [16] TBA yöntemi ile 181 örnek uzunluklu kalp işaretlerinden elde ettikleri 14 öznitelik ile gizli katmanında sırası ile 5, 15, 25 ve 30 nöron olan MLP YSA'ları ile PVC sınıflandırması yapmışlardır. 14x5x1 boyutlu YSA %94,09 doğrulukta sınıflandırma yaparken gizli katmanında daha fazla nöron bulunan 14x15x1 boyutlu başka bir ağ ile %93,76 doğrulukla sınıflandırma yapmıştır.

Jiang ve arkadaşları [18] ektopik ventriküler atımları %98,1 doğrulukta tanıyan, FPGA tabanlı bir aritmi sınıflandırıcı önermişlerdir. Bu sınıflandırıcı öznetelik çıkarım yöntemi olarak Hermit açılımını kullanmaktadır ve 14 adet hesaplama bloğuna sahiptir. Her bir hesaplama bloğu komşuluğunda bulunan diğer dört hesaplama bloğuna bağlanmıştır. Temel anlamda her bir blok 4 temel MLP nöronunun işlevselliğine sahiptir. $7 \times 2 \times 7$ boyutlu bu YSA mimarisi 56 işlemci elemana eşdeğer bir donanım özelliğindedir. Oluşturdukları yapının büyüklüğünün FPGA'nın donanım yoğunluğu ile sınırlı olduğu ve küçük ağlarda hesaplama zamanı kısaldığı için ağdaki işlemci elemanlarının sayısının azaltılmasına çalışmışlardır. Literatürde aritmi sınıflandırıcıların donanımsal olarak gerçekleştirilmesi üzerine yapılmış çalışmalar çok sınırlıdır. Jiang ve arkadaşlarının önerdiği bu donanım, ektopik ventriküler atımların YSA tabanlı bir sınıflandırıcı modeli ile FPGA yongası üzerinde oluşturduğu tek çalışma olarak bilinmektedir.

1.2. Çalışmanın Amacı

Literatürde önerilen YSA tabanlı otomatik aritmi sınıflandırıcıları genellikle çok karmaşık yazılım tabanlı mimarilerdir. Bunlardan bazıları onlarca giriş işlemci elemanı ve çok sayıda gizli katmana sahiptirler [20, 34]. Bu derecede büyük ağlar, oluşan hesaplama zorluklarından dolayı çok güçlü işlemciler üzerinde bilgisayar bağımlı olarak gerçekleştirilmek durumunda kalmaktadırlar. Bu tez çalışmasında literatürdeki mevcut bilgisayar bağımlı aritmi sınıflandırıcı yazılım modelleri yerine, YSA tabanlı taşınabilir bir otomatik PVC aritmi sınıflandırıcının Alan Programlanabilir Kapı Dizileri (FPGA, Field Programmable Gate Arrays) üzerinde donanımsal olarak gerçekleştirilmesi amaçlanmaktadır.

Sayısal, analog ve karma yöntemler kullanılarak, Çok Büyük Ölçekli Tümleşik Devreler (VLSI, Very Large Scale Integration) veya Uygulamaya Özel Tümleşik Devreler (ASIC, Application Spesific Integrated Circuit) üzerinde oluşturulmuş YSA donanımları bulunmaktadır [37]. Analog YSA gerçeklemeleri, düşük maliyet ve yüksek hız gibi avantajlarına rağmen zayıf hata bağışıklıkları ve sabit mimarileri büyük birer dezavantajdır. Diğer taraftan sayısal YSA gerçeklemeleri ise çok fazla donanım kaynağı kullanan büyük ölçekli çarpıcılar ve Doğrusal Olmayan Aktivasyon Fonksiyonu (NAF, Nonlinear Activation Function) bloklarına ihtiyaç duyarlar. Bu sebeplerden dolayı

genellikle YSA'lar, tek bir çarpıcı ve NAF bloğunun paylaşarak kullanıldığı pahalı bir mikroişlemci veya Sayısal İşaret İşlemciler (DSP, Digital Signal Processors) üzerinde gerçekleştirilirler.

Bununla birlikte mikroişlemciler ve DSP'lerin sıralı işlem özelliği, YSA'nın paralel mimarisinin avantajlarının kullanılabilmesi için uygun bir çözüm sunamaz. VLSI ve ASIC seçenekleri ise gerek çok uzun tasarım süreleri ve çok pahalı ilk üretim maliyetleri, gerekse sahip oldukları sınırlı esnekleri gibi dezavantajları yüzünden YSA gerçeklemeleri için çok uygun çözümler değildir. FPGA gerçeklemeleri ise tasarımcıya düşük maliyet, hızlı ilk örnekleme, uyarlanabilir donanım, güçlü yazılım geliştirme araçları desteği ve dağıtılmış paralel mimari gibi çok önemli avantajlar sunmaktadır [38].

FPGA tabanlı YSA gerçeklemelerinin tasarımcıya sunduğu avantajlara rağmen paralel YSA mimarilerinin ihtiyaç duyduğu çok fazla sayıda donanım kaynağı, YSA'ların FPGA üzerinde donanımsal olarak gerçekleştirilmesi uygulamalarında halen büyük bir problemdir. YSA tabanlı bir otomatik PVC aritmi sınıflandırıcıyı, tek bir FPGA yongası üzerinde gerçeklemek için YSA girişine uygulanan EKG işareti öznitelik sayısının azaltılması gerekir. Öznitelik sayısının azaltılması YSA giriş işlemci eleman sayısını ve buna bağlı olarak ağdaki eleman sayısını azaltacaktır. Bu sayede YSA donanımının ihtiyaç duyacağı donanım kaynak sayısı azaltılacak ve yonga alanında tasarruf sağlanacaktır. Bu tez çalışmasında YSA girişlerine uygulanan öznitelik sayısının azaltılması amacı ile MIT-BIH EKG veritabanından alınan, dört hastaya ait yarım saatlik EKG kayıtlarına TBA yöntemi uygulanması hedeflenmiştir. Her bir hastaya ait EKG kaydı değişik sınıflardan binlerce kalp atımından oluşmaktadır. TBA yöntemi ile orijinal veri daha az sayıdaki öznitelik ile ifade edilebilir ve bu sayede YSA'nın giriş sayısı azaltılabilir. Böylelikle FPGA üzerinde oluşturulacak olan YSA donanımının ihtiyaç duyacağı donanım kaynak sayısı da azaltılabilir.

Matlab ortamında TBA yöntemi ile boyutları azaltılmış olan YSA modelleri, daha az yonga alanı kullanan donanımlar olarak FPGA'lar üzerinde gerçekleştirilebilirler. Donanım kaynak kullanımlarını daha da azaltmak amacı ile donanım tasarımlarında kullanılan nümerik tanımlama formatları da değiştirilebilir. Bu tez çalışmasında yüksek hassasiyetli IEEE 754 32 bit kayan noktalı nümerik tanımlama ve ikili aritmetiği

kullanan 16 bit sabit noktalı nümerik tanımlama formatları kullanılarak YSA donanım gerçeklemeleri yapılması hedeflenmiştir. Bu yolla 16 bit sabit noktalı nümerik sayı sisteminin basit yapısı ve hesaplama kolaylıklarından faydalanılarak, FPGA üzerinde oluşturulacak olan YSA donanımının kaynak kullanımının daha da azaltılması sağlanacaktır.

Bu çalışma ile otomatik PVC aritmi sınıflandırıcılar için hızlı ve düşük maliyetli YSA tabanlı sınıflandırıcı donanımlarının ortaya konulması amaçlanmıştır. Oluşturulacak YSA donanımlarının tümüyle paralel ve yüksek hızlarda çalışması hedeflenmektedir. Her ne kadar EKG işaretleri düşük frekanslı işaretler de olsalar, uzun holter kayıtlarının hızlı bir şekilde değerlendirilmesi için bu tez çalışmasına önerilen donanımlar gibi hızlı karar birimlerine ihtiyaç duyulmaktadır. Bu tez çalışmanın amacı literatürdeki mevcut yazılım tabanlı YSA modellerinin yerine, klinikte aktif bir şekilde kullanılabilecek, düşük maliyetli ve hızlı otomatik PVC aritmi sınıflandırıcı donanımları ortaya koymaktır.

Kalbin çalışması ve EKG kayıtları hakkında kısa bilgilerin bulunduğu 2. Bölüm’de bu çalışmada kullanılan aritmi tipleri ve özellikleri verilmiştir. Ayrıca bu bölümde PhysioNet ve MIT-BIH EKG veritabanları tanıtılmış ve kayıtların özellikleri anlatılmıştır. Yine bu bölümde kayıtların okunması ve işlenmesinde kullanılan açık kaynak kodlu yazılımlar hakkında da kısa bilgiler bulunmaktadır.

3. Bölüm’de EKG işaretlerinin sınıflandırılmasında kullanılan en popüler öznelik çıkarım algoritmaları incelenmiş ve bu algoritmalar hakkında bilgiler verilmiştir. Bu yöntemlerden iyi sonuçlar üreten ve yaygın olarak kullanılanlardan birisi olan TBA yöntemi, bu bölümde detaylandırılmış ve yöntemin EKG kayıtlarına nasıl uygulandığı tarif edilmiştir.

Medikal tanı sistemlerinde ve daha birçok alanda yaygın olarak kullanılan bir sınıflandırma yöntemi olan YSA, 4. Bölüm’de anlatılmıştır. Bu bölümde YSA’nın, bu tez çalışmasında çözülmek istenilen probleme uygunluğu tartışılmış ve yöntemin hangi alanlarda nasıl kullanıldığı hakkında bilgiler verilmiştir. Bu bölümde YSA’nın iç bileşenleri, aktivasyon fonksiyonları ve öğrenme algoritmaları da tartışılmıştır.

Bu tez çalışmasında kullanılmak üzere, MIT-BIH veritabanından indirilmiş olan EKG kayıtlarının özellikleri ve hangi ölçütlerden dolayı bu kayıtların seçildikleri hakkında geniş bilgi 5. Bölüm’de verilmiştir. Bu bölümde tez çalışmasında kullanılan EKG kayıtlarına TBA yönteminin nasıl uygulandığı ve TBA uygulanan EKG işaretleri ile boyutları azaltılmış öznitelik setlerinin nasıl elde edildikleri gösterilmiştir. Boyutları azaltılmış bu öznitelik setleri ile eğitilmiş değişik boyutlu birçok ağın, farklı öğrenme algoritmaları ve aktivasyon fonksiyonlarına nasıl cevaplar ürettiği incelenmiştir. Ağ boyutu, öğrenme algoritması, aktivasyon fonksiyonu tipi ve ağ parametrelerinin nasıl belirlendiği bu bölümde anlatılmıştır.

İlgili YSA modelinin FPGA donanımın oluşturulması 6. Bölüm’de verilmiştir. Bu bölümde FPGA hakkında bilgiler verilmiş ve bu yonga teknolojisinin, YSA geliştirme uygulamaları için ne kadar uygun olduğu anlatılmıştır. YSA’nın doğasındaki paralelliğin ve FPGA yongalarının bağımsız işlem birimlerinin paralel çalışmayı nasıl mümkün kıldığı ifade edilmiştir. Nümerik tanımlama ve sayı formatları hakkında bilgiler verildikten sonra, FPGA ile donanım geliştirmede nümerik tanımlamanın ne derece önemli olduğu anlatılmış ve YSA modeli iki farklı nümerik tanımlama ile gerçekleştirilmiştir. Oluşturulan donanımların özellikleri bu bölümde ayrıntılı bir biçimde verilmiştir.

Son olarak 7. Bölüm’de sonuçlar tartışılmış ve bu bölümün ardından ekler kısmında Matlab kodları verilmiştir.

2. BÖLÜM

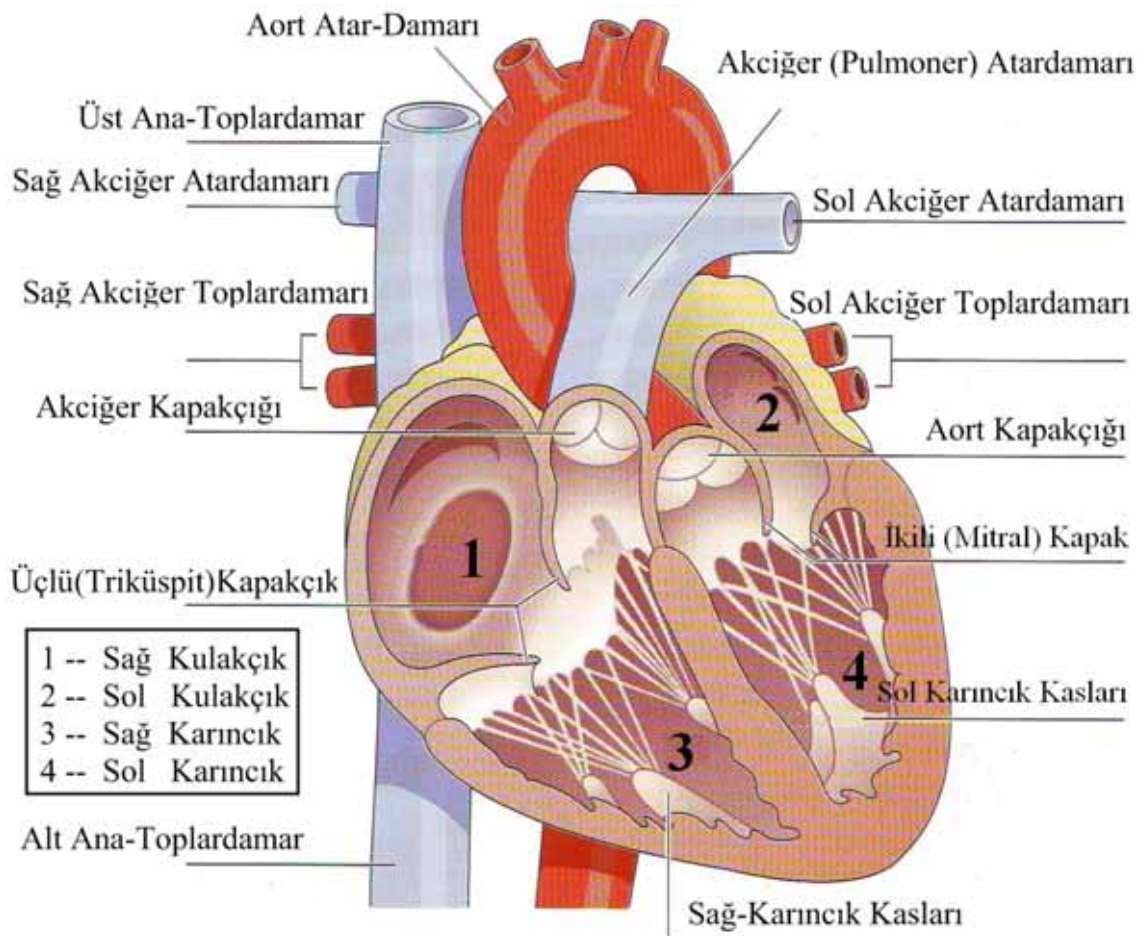
KALP VE ELEKTROKARDİYOGRAM

2.1. Kalbin Görevi ve Yapısı

Kalp vücuda düzenli olarak kan pompalanmasından sorumlu organdır. Düzenli olarak pompalanan kan, oksijen ve glikoz gibi hayati işlevleri olan maddelerin, beyin ve diğer bütün dokulara iletilmesini sağlarken, kanın kalbe geri dönüşü esnasında dokulardaki zararlı artık maddelerin arındırılması ve kanın oksijen ile zenginleştirilmesini de sağlar. Şekil 2.1’de görüldüğü gibi kalp, sağ/sol karıncık ve kulakçık olarak isimlendirilen dört odacıktan oluşur. Kalbin üst kısmında bulunan kulakçıklar dönen temiz veya kirli kanı toplar ve alt kısımda bulunan karıncıklara, kalp kapakçıkları üzerinden transfer eder. Karıncıklardaki basınç kulakçıklardaki basıncı geçtiği anda bu kapakçıklar kapanır ve kan akışı sonlandırılır; bu süreye diyastol denir. İçerisi kanla dolan karıncık, kasılarak içerisindeki kanı yüksek bir basınç ile dolaşım sistemine pompalar; bu süreye ise sistol denir. Koldaki atardamar üzerinden alınan tansiyon ölçümünde kullanılan büyük ve küçük tansiyonlar sırası ile ortalama 120 mmHg değerindeki sistol ve 80 mmHg değerindeki diyastol basınçlarıdır. Temiz kanı bütün vücuda, en uç dokulara kadar taşıyan büyük dolaşım sistemini besleyen kalbin sol kesiti, daha yüksek basınçla çalıştığı için sol karıncık hem hacim olarak hem de kas kütlesi olarak sağ karıncıktan daha büyüktür.

Akciğerlerden gelen temiz kan, akciğer (Pulmoner) toplardamarından sol kulakçığa taşınır ve sol artiyum kaslarının kasılması ile ikili (mitral) kapakçık üzerinden sol karıncığa transfer edilir. Sol ventrikül kaslarının kasılması ile mitral kapakçık kapanır ve karıncıktaki temiz kan artan sistol basıncı ile aort atardamarı üzerinden büyük dolaşım sistemine pompalanır. Sindirim sistemi, diğer organlar ve en uç dokulara kadar taşınan kan, üst ana toplardamar, alt ana toplardamar ve koroner sinüs ile sağ kulakçığa toplanır. Sağ artiyum kasları kasılarak üçlü (triküspid) kalp kapakçığını açar ve

içerisindeki kirli kanı sağ karıncığa aktarır. Sağ ventrikül kaslarının kasılmasıyla üçlü kapakçık kapanır ve kirli kan, akciğer atardamarı üzerinden akciğerlere pompalanır. Akciğerde temizlenen kan sol kulakçığa gelerek çevrimi devam ettirir. Sağ ve sol bölgedeki atriyum ve ventrikül kas hareketleri eş zamanlı olarak meydana gelir. Bunun anlamı, sağ ve sol kulakçıklar aynı anda kasılarak karıncıkların kanla dolmasını sağlarlar. Daha sonra, kanla dolan sağ ve sol karıncıklar da aynı anda sırası ile küçük (akciğer) ve büyük dolaşıma kan pompalarlar.



Şekil 2.1. Kalbin içyapısı ve önemli damarlar [39].

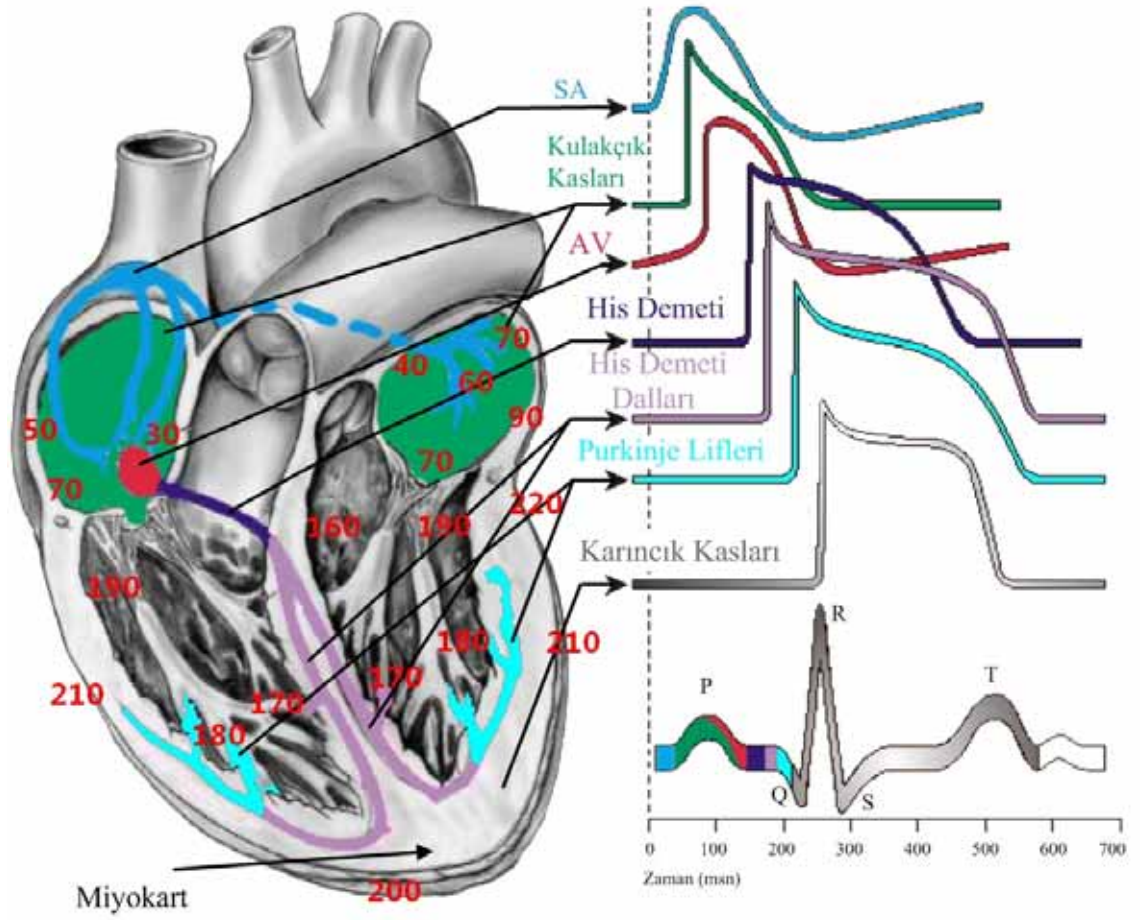
2.2. Kalbin Elektriksel İletim Sistemi

Miyokart (myocardium) kalp kasına verilen genel isimdir. Miyokart düzenli olarak kasılarak vücutta düzenli olarak kan akışını sağlar. Miyokardın düzenli olarak kasılmasını sağlayan aksiyon potansiyeli pace-maker yani uyarı-odağı tarafından üretilir. Pace-maker, sinüs düğümü olarak da adlandırılan SinoAtriyal (SA) düğümdür. Bazı kalp hastalarında bu düğümün işlevselliğinde sorunlar olduğu için kalp-pili

olarak adlandırılan pace-maker cihazları kullanır. 1950’li yıllardan beri kullanılan kalp-pilleri ritmik düzeni sağlamaktan sorumludur. Eski modelleri sürekli olarak elektriksel uyarılar üretirken, günümüzde kullanılan ve 5. nesil olarak adlandırılan kalp-pilleri, ritim bozuklukları olduğu zaman devreye girerek uyarı üretirler. Bu sayede hem pace-maker normal QRS’leri bozmamakta hem de daha uzun batarya ömrü sunmaktadır.

Normal bir kalp vurusu, sağ kulakçığın arka duvarında yer alan ve merkezi sinir sistemi tarafından kontrol edilen SA düğümünde oluşturulan, 20 mV genlik değerine kadar ulaşan aktivasyon potansiyeli ile başlatılır. SA düğümünde oluşturulan uyarı, depolarizasyon dalgası şeklinde kalbin düşük direnç bölgeleri olarak ifade edilen, Şekil 2.2’de gösterilen kalp iletim sistemi üzerindeki yolları izleyerek sonlanır. Kalp atım hızı sağlıklı bir insanda durgun vaziyette 60-100 atım/dakikadır ve bu değer SA düğümünün uyarı frekansına eşittir [40]. Kalp bir sonraki uyarı sinyali uygulanana kadar olan sürede dinlenir.

SA düğümünde oluşturulan uyarının kalbin iletim yollarındaki hızı yüksek olduğu için, pace vurusu özelleşmiş hücrelerden oluşan AtrioVentriküler (AV) düğüm tarafından geciktirilerek iletilir. Bu gecikmeler kanın, kulakçık ve karıncıklara yeterli miktarda dolması için bekleme sağlar. SA düğümü kulakçık kasılmasını başlatarak uyarıyı AV düğümüne ulaştırır. AV düğümü karıncıkların kanla dolmasına olanak veren fizyolojik gecikmeyi sağlar ve uyarıyı his demeti üzerinden sağ ve sol dallar yoluyla miyokarda taşıyarak karıncık kasılmasını başlatır. Purkinje lifleri kalp iletim sisteminin en hızlı bileşenleridir ve karıncıkların düzenli olarak depolarize olmasını sağlar. SA düğümü tarafından oluşturulan uyarının kalbe yayılması esnasında, değişik kısımlara uğrama süreleri milisaniye cinsinden Şekil 2.2’de verilmiştir [41]. Miyokardın kasılmasını sağlayan bu elektrik akımı, her bir normal atım oluşmadan önce belirtilen sıra ile kalp iletim yolu üzerinden geçer.



Şekil 2.2. Kalbin iletim sistemi ve uyarının yayılımı.

2.3. Elektrokardiyogram (EKG)

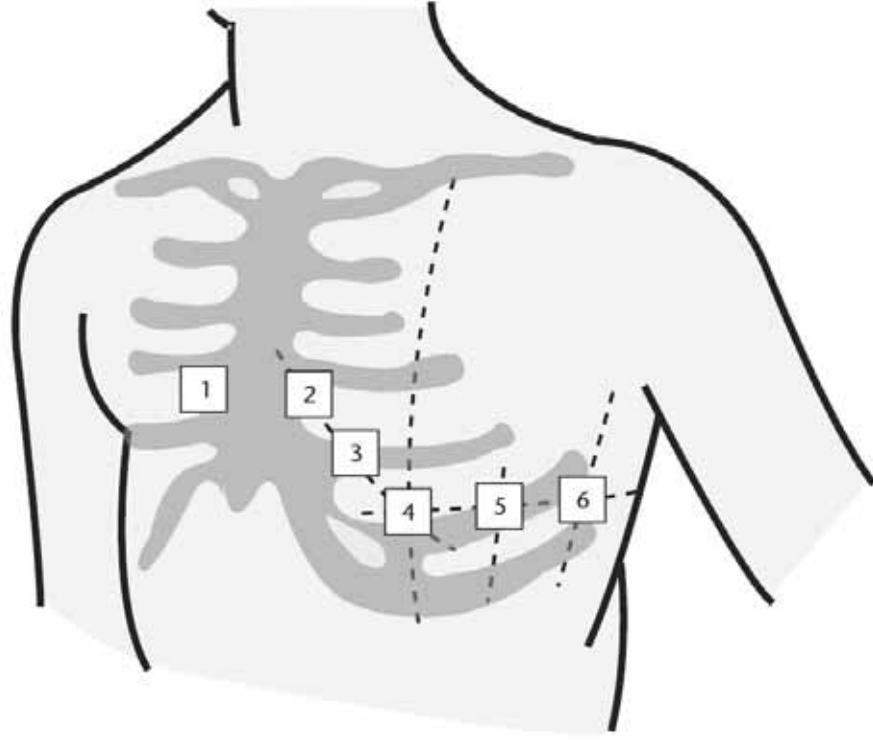
Kalbin vücuda düzenli olarak kan pompalaması amacıyla, miyokardın ritmik kasılmalarını sağlayan ve kalbi boydan boya geçen elektriksel uyarılar rastgele değildir. Her bir normal atım veya sistolden önce, SA düğümünde oluşturulan ve etkili bir kasılmanın sağlanması amacı ile kalbin içyapılarında koordine edilen uyarının örüntüsü Şekil 2.2’de verilmiştir. Kalp içerisindeki bu elektriksel aktivite, vücut yüzeyinde ölçülebilir elektriksel potansiyel farklar şeklinde bir etki oluşturmaktadır. Bu etkinin kuvvetlendirilmiş ve genellikle filtrelenmiş sinyal haline EKG denilmektedir [42]. Kalpteki iletim sistemi bozuklukları, kardiyak kasılmayı sağlayan kas liflerindeki anormallikler, miyokardtaki metabolizma bozuklukları veya kalpteki şekil bozuklukları gibi birçok etki EKG işaretinde değişimlere sebep olmaktadır. EKG analizi kalp rahatsızlıkları ile ilgili temel ve güvenilir klinik verisi olarak kabul görmektedir. Kalpteki elektriksel anormallikler EKG kayıtları ile tanımlanır.

2.3.1. EKG'nin Tarihçesi [43]

Kalbin kan pompalarken vücut yüzeyinde oluşturduğu ses, titreşim ve elektriksel etkiler tarih boyunca araştırmacıların ilgisini çekmiştir. Basit ve tümüyle açıklanabilir bir örüntüye sahip olması sebebiyle, bir medikal tanı sistemi olarak, EKG analizinin kullanımı oldukça eski bir geçmişe sahiptir. EKG'yi rutin klinik tanı sistemi haline getiren sürecin, kardiyak kas kasılmasının elektriksel ölçümünü mümkün kılacak galvonometre cihazının, 1786 yılında İtalyan bilim adamı Luigi Galvani tarafından icat edilmesi ile başladığı söylenebilir. 1895 yılında Einthoven ve Ueber “Des Menshilchen Electrocardiogramms” isimli kitap ile P, Q, R, S ve T tanımlamasını yapmış ve standart EKG ölçümü için elektrotların doğru bağlantısı için kullanılan Einthoven üçgenini tanımlamışlardır. Einthoven'ın geliştirmiş olduğu bir düzenek yardımı ile 1901 yılında yapmış olduğu ölçüm, ilk EKG ölçümü olarak kabul edilir. 1931 yılında Charles Wolferth ve Francis Wood EKG işaretindeki değişimlerin kalp krizi ile ilgisini incelemişlerdir. 1934 yılında Frank Wilson'un sağ kol, sol kol ve sol bacağı 5 Kohm'luk dirençler ile bağladığı iletkenlerle yaptığı ölçüm tekniği “Wilson Central Terminal” olarak anılmaktadır. Geliştirilmiş standart derivasyonlar olan aVR, aVL ve aVF derivasyonlarının tanımlaması da buradan gelmektedir. 1939 yılında Langendorf, atrial fibrilasyonu EKG işaretine bağlı olarak tanımlamıştır. İlk EKG kayıt cihazı ise tıp eğitimi almış olduğu halde mühendislik yönü ağır basan, İsveçli mühendis Rune Elmqvist tarafından tasarlanmıştır. 1950'li yıllar ile artık EKG işaretlerinin analizi teknikleri üzerine çalışmalar başlamıştır [43].

2.3.2. EKG'nin Elektriksel Özellikleri ve Ölçümü

SA düğümünde başlatılan uyarının kalbin iletim yolu üzerinden taşınması ile meydana gelen etki, Şekil 2.3'te gösterildiği gibi deri üzerinde çeşitli yerlere yerleştirilen elektrotlar ile algılanmaktadır. 1-2 mV genlikli EKG işaretinin frekans içeriği tipik olarak 0,5-100 Hz arasındadır. Ancak bu sinyalin üzerine birer bozucu etki olarak, frekans içerikleri 5 kHz'e kadar çıkan, kasların elektriksel aktivitesine karşılık gelen ElektroMiyogram (EMG) sinyali, elektrotların titreşimleri ile meydana gelen mekanik parazitler ve 50 Hz şebeke gürültüsü eklenmektedir [4]. Bu yüzden genellikle kayıt esnasında, EKG işareti bir ön filtreleme işlemine tabi tutularak istenmeyen bu frekans bileşenlerinden arındırılır. Herhangi bir filtreleme işlemi uygulanmadan kaydedilmiş EKG verileri ise yorumlanmadan önce bir filtreleme işlemine tabi tutulurlar [42].



Şekil 2.3. Elektrotların deri üzerindeki yerleşimi [42].

EKG ölçümlerinde bir standardın oluşturulması için belirli bir sıra ile göğüs kafesi hizasına yerleştirilen elektrotlardan, farklı kombinasyonlarda alınan potansiyel farklar derivasyonlar ile tanımlanmıştır. Klinikte altı adet ekstremité derivasyonu ve altı adet prekordiyal derivasyon olmak üzere, standart toplam 12 derivasyon mevcuttur. Prekordiyal (göğüs) derivasyonlar Şekil 2.3'te gösterilen elektrotlardan alınır ve V1, V2, V3, V4, V5, V6 derivasyonları olarak ifade edilir. Ekstremité derivasyonları ise ML I (D I), ML II (D II), ML III (D III) ve aVF, aVL, aVR olarak isimlendirilen, kollar ve sol bacdaktan alınan ölçümlerin de bileşkeye etkisi olan derivasyonlardır.

2.4. Kalpteki Ritim Bozuklukları

En belirgin kalp işlev bozukluklarının bir kısmı kalbin ritminin bozulması ile ortaya çıkar. Uyarı odağının SA düğümünden başka bir yere kayması, uyarının iletim yolu üzerinde herhangi bir yerde bloke olması, uyarının anormal bir yoldan ilerlemesi, kalbin herhangi bir yerinde kendiliğinden uyarıların oluşması veya fizyolojik sebeplerden biri veya birkaçının birleşimi, kalpte ritim bozukluklarına sebep olur.

Kalbin hızlanması, yavaşlaması veya düzensiz atımlarla çalışması gibi değişik özellikler gösteren ritim bozuklukları hakkında geniş bilgiye [41] numaralı kaynaktan ulaşılabilir.

2.4.1. Normal Sinüs Ritmi

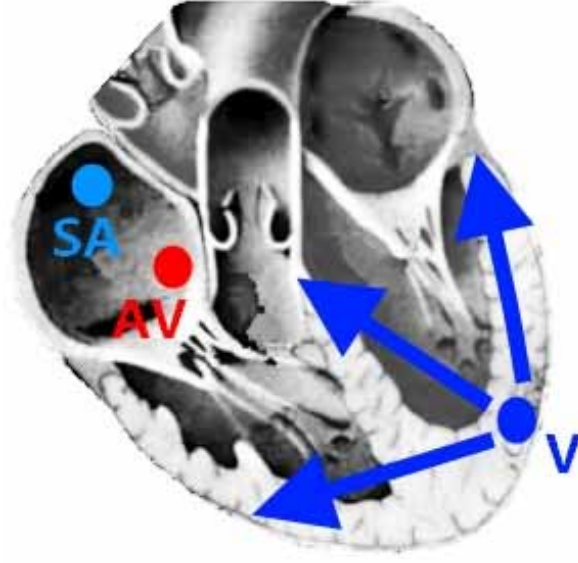
Normal bir kalp atımı, SA düğümünden başlatılan uyarının kalp iletim sistemi üzerinden doğru bir zamanlama ile miyokarda ulaştırılması ile tamamlanır. Sağlıklı bir insanın EKG kaydı normal ritimlerden oluşur. Normal sinüs ritmi Şekil 2.4'te, normal bir EKG işareti ise Şekil 1.1'de görülmektedir. EKG işaretinin bileşenleri kardiyak kasılmanın değişik evrelerini ifade eder. P dalgası kulakçık kasılmasına, P-R aralığı AV düğümdeki fizyolojik gecikmeye, QRS bileşke vurusu karıncık kasılmasına ve T dalgası karıncık repolarizasyonuna karşılık gelir [40]. Kulakçık repolarizasyonu yüksek genlikli QRS bileşke vurusu içerisinde kaybolduğundan EKG'de görünmez.



Şekil 2.4. Normal sinüs ritmi.

2.4.2. Erken Karıncık Kasılması (PVC, Premature Ventricular Contraction)

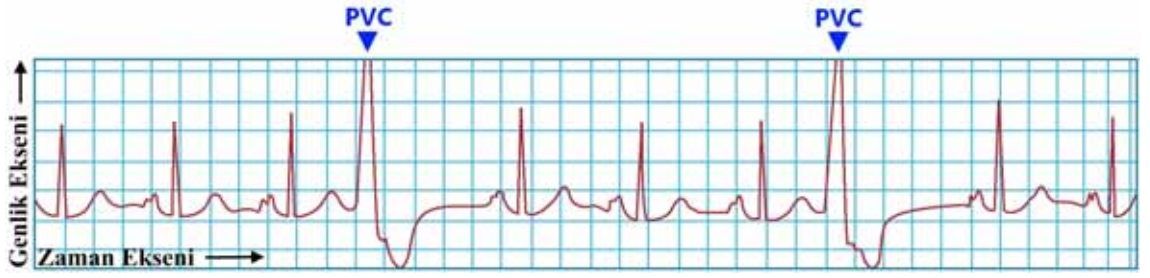
Erken Karıncık Kasılması (PVC, Premature Ventricular Contraction) uyarı odağının SA düğümünden başka bir odağa kayması sonucunda oluşur. Sağ veya sol karıncıkta, genellikle his-purkinje liflerinin yanındaki tek bir odak hızlanır ve kontrolü SA düğümünden alır. Odak merkezi karıncığa kaydığı ve normal ritimden daha erken oluştuğu için erken karıncık kasılması ismini alır. Şekil 2.5'te PVC'ye sebep olan odak kayması gösterilmiştir. PVC'nin başlıca nedenleri elektrolitik anormallikler, dilate kardiyomiyopatiler, iskemi gibi kalp hasarları, ilaç etkileri veya miyokarttaki viral enfeksiyonlardır [4, 44]. İstatistikler PVC tipi aritmilerin, ani ölümcül VF'yi başlatma riskinin çok yüksek olduğunu göstermektedir [10]. Bu bakımdan erken tanı ile tedavi edilmeleri oldukça önemlidir.



Şekil 2.5. Erken karıncık kasılmasına sebep olan odak kayması.

PVC tipi aritmilerin tipik dalga karakteristiği anormal derecede geniş QRS ve T aralıklarına sahip olmalarıdır. 0,12 saniyeden uzun ve genellikle yüksek genlikli bir QRS aralığına sahiptir. Bunun sebebi odağı kaymış uyardının, his-purkinje sistemi değil de daha çok iletimi yavaş olan karıncık kası ile iletilmesidir. QRS dalgasındaki genlik yükselmesinin sebebi ise yine karıncığın herhangi bir noktasına kaymış vuru odağından kaynaklıdır. Normal vuru SA'dan başlatılarak his-purkinje üzerinden eşit sürelerde miyokarda düzenli bir şekilde ulaştırılır ve zıt yönlerde oluşan depolarizasyon dalgaları birbirlerini nötralize eder. Fakat odak kayması sonucu oluşan uyarı, Şekil 2.4'te görüldüğü gibi tek yönde hareket ettiği için, kalbin bir yanı diğerinden daha hızlı kasılır ve nötralize etkisi oluşmaz. Hemen hemen bütün PVC vurularında QRS'i izleyen T dalgası, QRS potansiyeli ile ters yönlüdür [41].

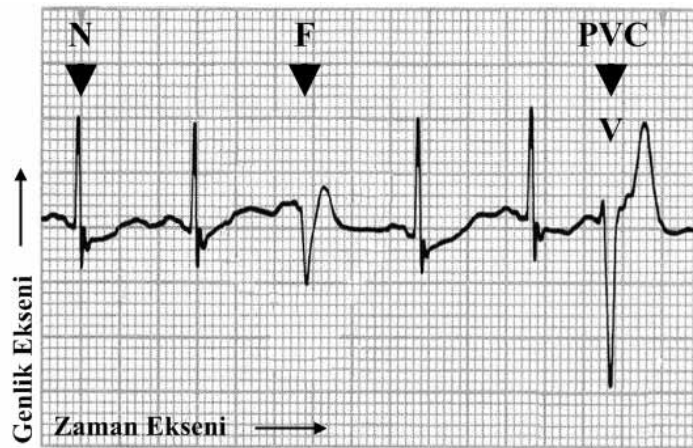
PVC tekrarlı atımlardan oluşmaz, genellikle bir normal atımdan önce veya sonra oluşur. Bir EKG kayıt kağıdında 60-100 vuru/saniye normal atımı için, bir satır boyunca sırası ile 12-15 atım görüntülenir. Bir satır boyuca 6 adetten fazla PVC görüntülendiğinde tehlikeli olarak nitelendirilir. PVC tipi vurular genellikle düzensiz görülür ve bazı durumlarda bir haftalık holter kayıtları incelenerek tanı konulmaya çalışılabilir [5]. Şekil 2.6'da bir EKG kaydı içerisinde yer alan PVC atımları gösterilmiştir.



Şekil 2.6. Erken karıncık kasılması [44].

2.4.3. Füzyon (Fusion) Aritmisi

Füzyon, kendiliğinden oluşan bir depolarizasyon sonucunda karıncıklarda meydana gelen zamansız bir aktivitedir. İki farklı uyarı odağı üzerinden gelen uyarının kasılmaya etkisi olarak da tanımlanabilir. Bu uyarı odakları genellikle SA düğümü ve karıncık üzerindeki ektopik bir odaktır. Bu odaklar aynı anda fakat farklı noktalardan, karıncık miyokardını uyarırsa füzyon akımı oluşabilir. Ektopik bir kulakçık odağı ya da AV düğüm bir karıncık odağı ile füzyon oluşturabilir. Benzer şekilde iki farklı karıncık odağı uyarı üreterek füzyon akımı oluşturabilir. Bu durum genellikle birden fazla uyarı odağı aktifken ortaya çıkar [45]. Bu tez çalışmasındaki füzyon atımlar, PVC aritmisine sebep olan uyarı odağının oluşturmuş olduğu uyarının, SA odağından çıkan normal uyarı ile çakışması sonucu karıncıkta oluşturduğu füzyondur. Füzyon, PVC aritmide olduğu gibi odağın yerine göre EKG işareti üzerinde değişik şekillerde görülebilir. PVC atımda olduğu gibi geniş bir QRS aralığına sahiptir. Şekil 2.7’de bir füzyon atım, normal atımlar ve bir PVC atımı ile birlikte gösterilmektedir.



Şekil 2.7. Füzyon (F) atımı.

Bu tez çalışmasında Normal (N), PVC (V) ve Füzyon (F) tipi üç grup aritmi için sınıflandırma yapılmıştır. Diğer birçok aritmi tipi için detaylı bilgiye [40-42, 44] numaralı kaynaklardan ulaşılabilir.

2.5. MIT-BIH Aritmi Veritabanı

MIT-BIH (Massachusetts Institute of Technology – Belt Israel Hospital) aritmi veritabanı [35] her biri 30 dakika uzunluğunda 48 adet çift kanal EKG kaydından oluşmaktadır. Bu kanallar A ve B olarak isimlendirilirler. A kanalı, kayıtların 45 tanesinde MLII derivasyonunu temsil ederken kalan üç kayıta V5 derivasyonunu temsil eder. B kanalı, 40 kayıta V1 derivasyonu olarak, kalan 18 kayıta ise MLII, V2, V4 ve V5 derivasyonları olarak seçilmiştir. 23 kayıt rutin klinik EKG verisi bulundururken diğer 25 kayıta özel aritmiler bulunmaktadır [5]. Medikal Enstrümantasyonda Gelişmeler Derneği'nin (AAMI, Association for the Advancement of Medical Instrumentation) [46] aritmilerin ifade edilmesinde önermiş olduğu standart ve MIT-BIH kayıtlarının içeriği Tablo 2.1'de verilmiştir.

Tablo 2.1. AAMI atım sınıfları ve MIT-BIH içeriği [5].

AAMI Atım Tipleri	N	S	V	F	Q
Tanımlama	S, V, F veya Q Dışındaki Atımlar	Supraventriküler Ektopik Atımlar	Ventriküler Ektopik Atımlar	Fusion Atımlar	Tanımlanamayan Atımlar
MIT-BIH Aritmi Tipleri	Normal				Paced (P)
	Atım (NOR)	Erken Kulakçık	Erken Karıncık	Normal ve	
	Sol Dal Bloğu	Kasılması (AP)	Kasılması (PVC)	karıncık atım	Paced ve Normal
	Atımı (LBBB)			füzyonu	Atım Füzyonu
	Sağ Dal Bloğu	Nodal Erken			(fPN)
	Atımı (RBBB)	Kasılma (NP)	Ventriküler		
	Atrial Kaçak		Kaçak (VE)		Sınıflandırılma-
	Atım (AE)	Supraventriküler			mış Atım (U)
	Nodal Kaçak	Erken Atım (SP)			
	Atım (NE)				

MIT-BIH veritabanındaki kayıtlar 0,1-100 Hz bant geçiren bir filtreden geçirildikten sonra 11 bit çözünürlükle, 10 mV'luk bir sahada, 360 örnek/saniye hızında örneklenmiş EKG verilerinden oluşur. Her bir hasta kaydı için üç adet dosya bulunur. Bunlar .dat uzantılı EKG kaydı, .atr uzantılı her bir atımın tipinin bağımsız iki kardiyolog tarafından işaretlendiği aritmi tanım dosyası ve .hea uzantılı hasta bilgilerinin (yaşı,

cinsiyeti, sađlık durumu ve EKG kaydının kısa bir tanımı) bulunduđu bir text dosyasıdır. Her bir kayıt binlerce atımdan meydana gelmektedir [35].

Bu tez çalışmasında MIT-BIH aritmi veritabanından 205, 208, 210 ve 213 numaralı kayıtlar kullanılmıştır. Bu dört kayıt için A kanalı ML II, B kanalı ise V1 derivasyonu şeklinde kaydedilmiştir. ML II derivasyonu kaydedilirken elektrokardiyografin negatif elektrodu sađ kola, pozitif elektrodu ise sol bacağına bağlanır. Sađ kol sol bacağına göre negatif duruma geçtiğinden, cihaz pozitif kayıt yapar. Bu tez çalışmasında elde edilen sonuçları, literatürdeki çalışmaların sonuçları ile kıyaslayabilmek için, 205, 208, 210 ve 213 numaralı kayıtların A kanalı yani ML II ekstremite derivasyonu kullanılmıştır [16]. Her bir derivasyon için, EKG cihazının ölçüm aldığı elektrotların bağlantı şekilleri farklıdır [40, 41].

Çalışmalarda MIT-BIH veritabanından alınan EKG kayıtlarının kullanımı, yapılan çalışmaların literatürdeki benzerleri ile karşılaştırılması adına önem taşımaktadır. Araştırmacılar algoritmalarının başarımlarını, diğ er araştırmacıların çalışmaları ile karşılaştırırken, genellikle daha önce yapılmış çalışmalarda kullanılan veri setleri ile çalışırlar. Ayrıca gerçek veriler ile çalışmanın bazı zorlukları bulunmaktadır. Bunlar; her türlü aritmi tipi için hasta bulmanın zorlukları ve gürültülü veya iyi ölçülememiş EKG kayıtlarının işlenmesinde karşılaşılan problemler olarak sıralanabilir. Bu tez çalışmasının literatürdeki çalışmalar ile karşılaştırılabilmesi ve geliştirme aşamasında çeşitli kolaylıklar sunması sebepleri ile EKG verileri MIT-BIH veritabanından alınarak kullanılmıştır.

2.6. PhysioNet

MIT-BIH veritabanının da bir parçası olduđu ve 50 farklı veritabanını barındıran PhysioNet [47], 700 GByte'ın üzerinde 10.000'den fazla fizyolojik sinyal kaydının olduđu PhysioBank ve açık kaynak kodlu geliştirme yazılımlarının bulunduđu PhysioToolkit isimli iki gruptan meydana gelir. PhysioToolkit, sinyal işlemede kullanılan pek çok programa ve algoritmaya açık kaynak kodlu olarak erişim imkanı sađlayan, Veritabanı Dalga Şekli Görüntüleme (WFDB, WaveForm DataBase) isimli bir kütüphaneye sahiptir. Bu kütüphanede kayıt dosyalarının Windows, Linux veya Matlab ortamlarında işlenmesini sađlayan birçok yazılım ve script bulunmaktadır.

EKG kayıtlarının izlenmesi ve üzerlerinde işlemlerin yapılabilmesi amacı ile Linux ortamında çalışan WAVE isimli bir program bulunmaktadır. Bu programın Windows ortamında çalışması için cygwin [48] programının bilgisayarda kurulu olması gerekmektedir. Bu program Windows altında Linux ortamı yaratarak, WAVE programının Windows işletim sistemine sahip olan bir bilgisayarda çalıştırılabilmesini sağlamaktadır. Şekil 2.8’de MIT-BIH veritabanından alınmış 100 numaralı kayıt WAVE programı ile görüntülenmiştir. Burada A kanalının ML II, B kanalının ise V5 derivasyonunu temsil ettiği ve 100 numaralı kaydın 0:18 ile 0:30 saniyeleri arasında Normal (N) atımlar halinde ilerlediği görülmektedir.



Şekil 2.8. WAVE programı ekran görüntüsü.

2.7. Veri Okuma Yöntemi

WFDB kütüphanesinde, EKG kayıtlarını bilgisayar ortamına aktarmak ve üzerlerinde işlemler yapmak için hazırlanmış komutlar bulunmaktadır. Bu tez çalışmasında hasta kayıtlarının Matlab ortamına aktarılması için rdsamp, kayıtlardaki aritmi tiplerinin okunması için rdann ve zaman aralıklarını saniye bilgisine çevirmek için kullanılan time2sec komutları WFDB kütüphanesinden alınmıştır. Bu kütüphanede kayıtların üzerine yazmak veya WFDB formatındaki kayıtların özellikleri hakkında bilgiler almak/değiştirmek amaçları ile altı farklı komut daha bulunmaktadır [47].

3. BÖLÜM

EKG İŞARETLERİNİ SINIFLANDIRMADA KULLANILAN ÖZNİTELİK ÇIKARIM ALGORİTMALARI

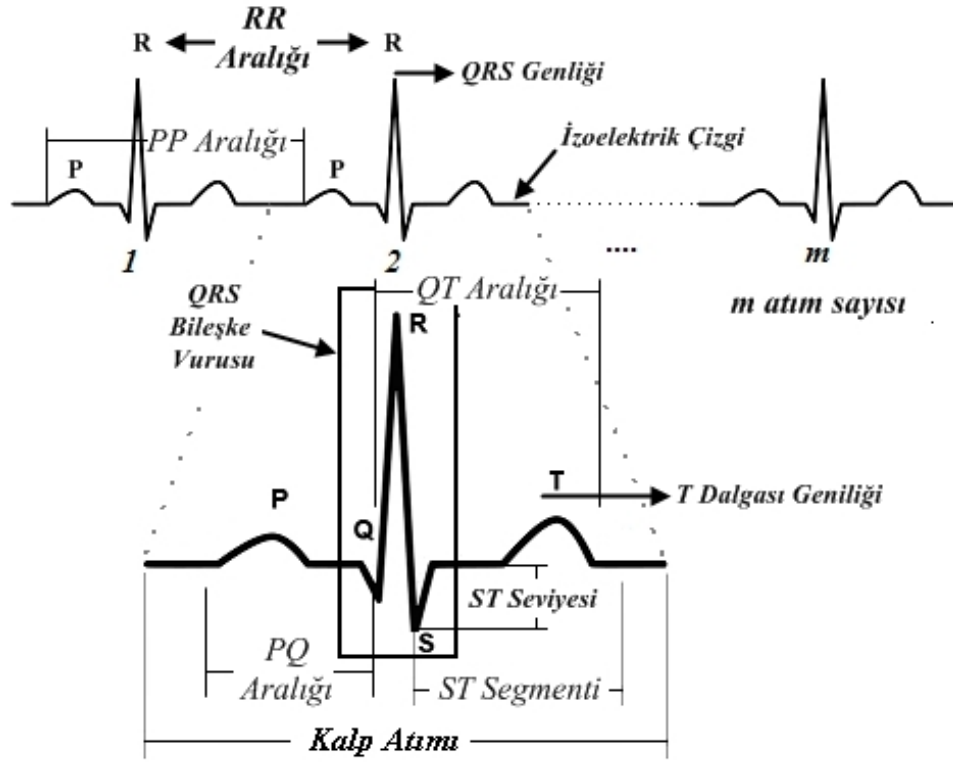
EKG işaret işleme uygulamalarında, kuvvetlendirilmiş ham EKG verisi bir ön filtreleme işlemine tabi tutulur. Bu yolla işaret istenmeyen bileşenlerinden arındırılır. Daha sonra çalışmanın odağını oluşturan sinyal karakteristiklerinin daha belirgin olarak elde edilmesi amacıyla uygun bir öznitelik çıkarım yöntemi EKG işaretine uygulanır. Öznitelik çıkarım yöntemleri, EKG verisini yüksek doğruluklarda az sayıda öznitelik ile tanımlamayı amaçlarlar. Öznitelik çıkarım işleminden sonra elde edilen bu öznitelikler uygun bir algoritma ile yorumlanarak sonuç çıkarımı sağlanır. Şekil 3.1’de bu süreç resmedilmiştir.



Şekil 3.1. EKG işaret işlemede izlenen adımlar.

Öznitelik çıkarım işleminde kullanılacak yöntemin sonuç çıkarım birimi ile uyumluluğu oldukça önemlidir. Bu yüzden öznitelik çıkarım biriminde kullanılan yöntem, sonuç çıkarım birimindeki çözümleme yöntemi ile doğrudan bağlantılıdır. EKG işareti işleme uygulamaları birçok alanda değişik şekillerde yapılmaktadır [4]. Bunlara örnek olarak EKG işaretinin biyometrik bir parametre olarak kullanıldığı uygulamalar [49-53], EKG işaretleri içerisindeki R tepelerini yakalayan çalışmalar [54, 55], veri sıkıştırma uygulamaları [54, 56, 57] ve çok yoğun bir şekilde EKG işaretlerinden aritmilerin tanındığı uygulamalar [5, 18, 21, 23, 24, 31, 34, 58, 59] verilebilir. Genel anlamda EKG ile ilgili olarak yapılan çalışmalar aritmi sınıflandırma uygulamaları ve diğer EKG işaret işleme uygulamaları şeklinde iki grupta halinde verilebilir.

EKG işaretleri ile ilgili tanımlar yapılırken aralık (interval), genlik, segment, kalp hızı ve QRS bileşke vurusu gibi terimler kullanılır. Bu terimlerin daha iyi anlaşılabilmesi için normal bir sinüs ritmi içerisinde, sık kullanılan tanımların ifade edildiği bir gösterim Şekil 3.2’de verilmiştir.



Şekil 3.2. EKG işaretleri ile ilgili bazı tanımlamalar.

Sağlıklı erkek erişkinlerde normal (sinüs) ritmi (60bpm/sn) boyunca gözlenen EKG özelliklerinin (V2) ortalama, minimum ve maksimum değerleri Tablo 3.1’de verilmiştir.

Tablo 3.1. Sağlıklı erişkin bir erkekte EKG özellikleri [42].

Özellikler	Normal Değerler	Normal Limitler
P Genişliği	110 ms	± 20 ms
PQ/PR Aralığı	160 ms	± 40 ms
QRS Genişliği	100 ms	± 20 ms
P Dalga Genliği	0,15 mV	$\pm 0,05$ mV
QRS Genliği	1,5 mV	$\pm 0,5$ mV
ST Seviyesi	0 mV	$\pm 0,1$ mV
T Dalga Genliği	0,3 mV	$\pm 0,2$ mV

Aritmi sınıflandırma uygulamalarında birçok öznitelik çıkarım tekniği kullanılmaktadır [4]. Aritmi çıkarım uygulamalarında kullanılan bütün öznitelik çıkarım yöntemlerini incelemek yerine, konunun daha özelleştirilmesi için bu tez çalışmasında odaklanılan PVC tipi aritmilerin tespiti uygulamalarında kullanılan öznitelik çıkarım teknikleri üzerine durulmuş ve birinci bölümde bu tekniklerin bir literatür özeti verilmiştir. PVC tipi aritmilerin sınıflandırılmasında kullanılan teknikler Kalp Atımı Aralıkları (Heart Beat Intervals) [5, 15, 18, 23, 24], frekans özellikleri [25, 29], Dalgacık Dönüşümü (DD) [14, 15, 19, 30, 31] ve Temel Bileşen Analizi (TBA, Principal Component Analysis) [16, 21, 32] olmak üzere dört farklı öznitelik çıkarım yöntemi olarak verilebilir.

3.1. Kalp Atımı Aralıkları ve Morfolojik Özellikler

Ardışık iki QRS vurusu arasını ifade eden kalp atım periyoduna eşit olan, R tepeleri arasındaki süreyi ifade eden R-R aralığı en yaygın kullanılan aralık (interval) tipidir. Çünkü PVC ve füzyon atımları kalpte farklı bir vuru odağının gelişmesi ile meydana gelir ve düzensiz olarak gözlenir. Bu yüzden R-R aralığı kalp atımlarındaki anormal değişimi gösterebilir. Ardışık kalp atımlarının bir önceki ve bir sonraki arasındaki R-R aralığı süresi, holter kaydındaki ortalama R-R aralığı süresi ile kıyaslanabilir. Fakat bu aralık tek başına PVC tipi aritmileri tanımlayamaz. Bu yüzden genellikle DD [15, 23] ve/ya ilave aralık ve/ya morfolojik [5, 24] ilave özellikler ile beraber kullanılarak, sınıflandırmanın doğruluğu arttırılmaya çalışılır. Uzun QRS dalga süresi PVC atımların en belirgin özelliğidir ve PVC sınıflandırmada, tanımlayıcı bir aralık olarak kullanılır [5]. Bunun yanında ST aralığı da [24] öznitelik seti içerisine eklenebilir.

Aralık özelliklerinin yanında morfolojik özellikler de sınıflandırma veri setleri içerisinde görülebilir. Örneğin QRS ve T dalga genliği [5] veya izoelektrik çizgi genliği [23] de öznitelik veri setleri içerisinde yer alabilir. PVC tipi atımlarda T dalgası R dalgası ile ters bir polarizasyona sahiptir. Eğer R pozitif polarizasyona sahipse, T negatif polarizasyona veya R negatif polarizasyona sahipse T pozitif polarizasyona sahiptir. Bu özelliğinden dolayı, T ile R dalgaları arasındaki bu ilişki, tanıya yardımcı önemli bir veridir.

PVC tipi aritmilerde uyarı odağının kayması sebebi ile orijinal dalga formunu kaybetmiş olan QRS bileşke vurusu, standart bir dalga örüntüsü oluşturamaz. Bu durum füzyon tipi atımlar için de geçerlidir. PVC tipi atım uyarı odağının yerine göre, R tepesinde Şekil 2.6'da görüldüğü gibi çok yüksek bir pozitif genlik oluşturabileceği gibi, depolarizasyon dalgasına ters yönde oluşan bir uyarı odağı Şekil 2.7'deki gibi negatif bir genlik meydana getirebilir. Bu yüzden PVC dalga karakteristiği mümkün olduğunca fazla sayıda aralık ve morfolojik özellik ile tanımlanmaya çalışılır. QRS işareti parçalara (segmentlere) bölünerek, tanıya yardımcı fizyolojik özellikleri çıkarılır ve bu özellikler bir sınıflandırma metodu ile anlamlı hale getirilmeye çalışılır. Aralık ve morfolojik özellikler kullanılarak, PVC aritmilerin sınıflandırmasını yapan değişik teknikler mevcuttur. Bu yöntemler başlıca YSA [15, 18] olmak üzere, HMM [23], Doğrusal Ayırtaç Fonksiyonu [24] ve istatistiksel [5] yöntemler olarak sıralanabilir.

Kalp atımı aralıkları ve morfolojik özellikler ile öznitelik seti oluşturmanın en büyük dezavantajı doğru bir sınıflama için gerekli olan öznitelik setinin çok detaylı olması gerekliliğidir [5]. Çok miktardaki veri hesaplama yükünü arttırır. Bu durum her ne kadar bilgisayar ortamında koşan yazılım tabanlı çözümler için önemli bir sorun teşkil etmese de gerçek zamanlı uygulamalarda büyük bir dezavantajdır. Mobil uygulamalarda ise hesaplama yükü çok ciddi bir dezavantajdır. Mobil cihazlar uygulamaya yönelik olduklarından ucuz ve düşük güç tüketimli olmalıdırlar. Hesaplama yükü güçlü işlemcilerin kullanımını gerektirir ve bu zorunluluk daha pahalı işlemcilerin kullanılmasını gerektirdiğinden maliyeti arttırır. Ayrıca güçlü işlemcilerin ihtiyaç duyduğu yüksek güç de ayrı bir dezavantaj olarak, mobil gerçekleştirilebilirliğini zorlaştırarak bu yöntemin bilgisayara bağımlı kalmasına sebep olmaktadır.

3.2. Frekans Özellikleri

EKG işaretindeki otomatiklik ve tekrarlanabilirlik özellikleri sebebi ile ritimdeki anormallikleri sezmek için EKG işaretinin frekans özellikleri kullanılabilir. Bu amaçla adaptif filtre [25] ve vektör-kardiyogram [29] yöntemleri ile sınıflandırma yapılmıştır. Buna rağmen PVC tipi otomatik aritmi sınıflandırıcılarda kullanılan en yaygın frekans analiz yöntemi DD'dir. Ancak DD yöntemi frekans bileşenlerinin yanında zaman bileşenlerini de kullandığı için ayrı bir başlık altında bir öznitelik çıkarım yöntemi olarak incelenmelidir.

3.3. Dalgacık Dönüşümü (DD)

Biyomedikal işaret işleme uygulamalarında frekans domeni bileşenleri birçok yöntem ile yorumlanırken, bunlardan en yaygın olanı Fourier Dönüşümüdür (FD, Fourier Transform). FD, işaretteki frekans domeni bilgileri hakkında çok detaylı sonuç çıkarımları yaparken, bunlara karşılık gelen zaman bilgilerini yoruma katamaz. Diğer taraftan Dalgacık Dönüşümü (DD, Wavelet Transform) durağan olmayan EKG işaretini frekans ve zaman bilgileri ile tanımlayabilir. Bu özelliği DD'yi işaret içerisindeki önemli faz bilgilerini korumak için uygun bir doğrusal operatör haline getirir [60]. Durağan olmayan işaretler için zaman ölçekli bir analiz yöntemi olan DD, PVC tipi aritmilerin sınıflandırılmasında kullanılan en yaygın öznitelik çıkarım yöntemidir ve genellikle Bulanık YSA (BYSA) ile birlikte kullanılırlar [14, 15, 19, 23, 30, 31].

Dalgacık dönüşümünde zaman tanımlı bir bölgede verilen işaret farklı ölçek ve çözünürlüklerde birçok bileşene ayrılır. DD'de amaç, işareti ölçek parametreleri ile etiketlenmiş başlıca fonksiyon kümeleri altında toplamaktır. Temel fonksiyonlar, dalgacık fonksiyonunun genleşmesi, büzülmesi ve kaydırılması ile işaretin ölçeklenmiş daha küçük yerel bölgelerinde analizine imkân verir. Genleşme uygulanan yerel bölgelerde düşük frekans bileşenleri, büzülme uygulanan yerel bölgelerde ise yüksek frekans bileşenleri incelenir.

$\varphi(t)$ sürekli bir işaret olsun, bu işarete dalgacık dönüşümü uygulanabilmesi için Denklem 3.1'deki şartın sağlanması gerekmektedir. Bunun anlamı $\varphi(t)$ sürekli işaretindeki salınımın alanı 0'a eşittir.

$$\int_{-\infty}^{\infty} \varphi(t) dt = 0 \quad (3.1)$$

Sürekli zamanda verilen bir işaret için Sürekli Dalgacık Dönüşümü (SDD) Denklem 3.2'de verildiği gibi bulunur. Bu dönüşüm enerji bileşenlerinin bulunması amacı ile yapılır ve orijinal işaret, dalgacık dönüşümünden tekrar elde edilebilir.

$$SDD(a, \tau) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t) \varphi^* \left(\frac{t - \tau}{a} \right) dt \quad (3.2)$$

Burada a ve τ sırasıyla ölçekleme faktörünü ve dalgacık fonksiyonunun zamandaki konumunu verir. $*$ karmaşık eşleniği temsil eder. Ölçekleme faktörünün büyütülüp küçültülmesi ile ana dalgacık fonksiyonunun genişmesi ve daralması sağlanır. Bu sayede $SDD(a, \tau)$ (dalgacık dönüşümü katsayıları) hesaplanır. Sürekli dalgacık dönüşümünde giriş sinyali ve (a, τ) ölçekleme-zaman parametreleri süreklidir. Bunun anlamı SDD her ölçekte ve bütün zaman bölgelerinde sürekli kaydırılarak gerçekleştirilebilir [61].

3.4. Temel Bileşen Analizi (TBA)

Temel Bileşen Analizi (TBA, Principal Component Analysis) PVC tipi aritmlerin sınıflandırılmasında kullanılan önemli bir öznitelik çıkarım yöntemidir [16, 21, 32]. EKG işaretlerinden öznitelik çıkarım uygulamalarında sıklıkla kullanılan güçlü ve güvenilir bir algoritmadır [34, 62]. TBA yöntemi güçlü genelleme yeteneğinden dolayı YSA tabanlı sınıflandırma uygulamalarında etkili sonuçların elde edilmesini mümkün kılmaktadır [34, 63].

TBA'nın üç temel amacı vardır.

- Verilerin boyutunu azaltmak,
- Tahmin yapmak,
- Veri setini bazı analizler için görüntülemek.

Genelleştirme yoluyla çalışan TBA yöntemi, veri indirgeme ve tahmin özelliklerine sahiptir. Bu özellikleri sebebiyle TBA yönteminin YSA ile uyumu oldukça yüksektir. Az sayıdaki veri ile işareti yüksek doğruluklarda tanımlayabildiği için çok az sayıda öznitelikle makul hatalar üreten yüksek yaklaşımlar sağlamaktadır. YSA'lar ile birlikte kullanıldıkları aritmi sınıflandırma uygulamalarında çok iyi sonuçlar üretmişlerdir. Düşük boyutlu verilerin uygulandığı sınıflandırma biriminin işlem yükü önemli ölçüde azalmakta, bu sayede güçlü işlemcilerin kullanımı bir zorunluluk olmaktan çıkmaktadır.

TBA orijinal veriyi mümkün olan en az sayıda örnekle ifade edebileceği, küçük bir veri seti ile tarif etmek için kullanılan bir öznitelik çıkarım metodudur. Genellikle veri analizlerinde, aralarında önemli ölçüde korelasyon olan verilerin aynı veri seti içerisinde bulunması arzu edilmez. Çünkü aralarında yüksek derecede ilişkinin bulunduğu veriler, gereksiz olarak veri setinin boyutunu büyütür ve işlem karmaşasına sebep olurlar [64, 65].

TBA aralarında yüksek derecede korelasyon bulunan p sayıda bileşeni daha az sayıda ($p > n$) n adet bağımsız bileşen ile ifade eder. Bu bağımsız bileşenlere Temel Bileşen (TB) adı verilir. TBA ham veri seti içerisindeki örüntüleri benzerlik ve farklılıkları ile birlikte yeni bir veri ekseninde ifade eder. TB'ler ile oluşturulan bu yeni veri ekseninde, TB'ler korelasyonu en düşük olandan en yükseğe doğru sıralanır. Bu yeni veri setinde ilk TB en belirleyici, son TB ise en genel nitelikli özniteliktir. Bu veri setinde değişkenliğin yüksek olması korelasyonun düşük olduğunu ifade eder. Böylece ham (orijinal) veri setinin daha az sayıda bağımsız bileşen ile ifade edilmesi sağlanır [64].

Temel bileşenlerin üç temel özelliği aşağıdaki gibi sıralanabilir.

- Korelasyonsuzdurlar,
- Birinci temel bileşen toplam değişkenliği en çok açıklayan değişkendir,
- İkinci temel bileşen kalan değişkenliği en çok açıklayan bileşen ve sırası ile gelen diğer bileşenler ise kendilerinden sonra kalan değişkenliği en çok tanımlayan bileşenlerdir.

EKG işaretlerinden öznitelik çıkarımı konusundaki yüksek performansından dolayı, bu tez çalışmasında öznitelik çıkarım algoritması olarak TBA yöntemi kullanılmıştır.

3.4.1. Tez Çalışmasında Kullanılan Kayıtların Özellikleri

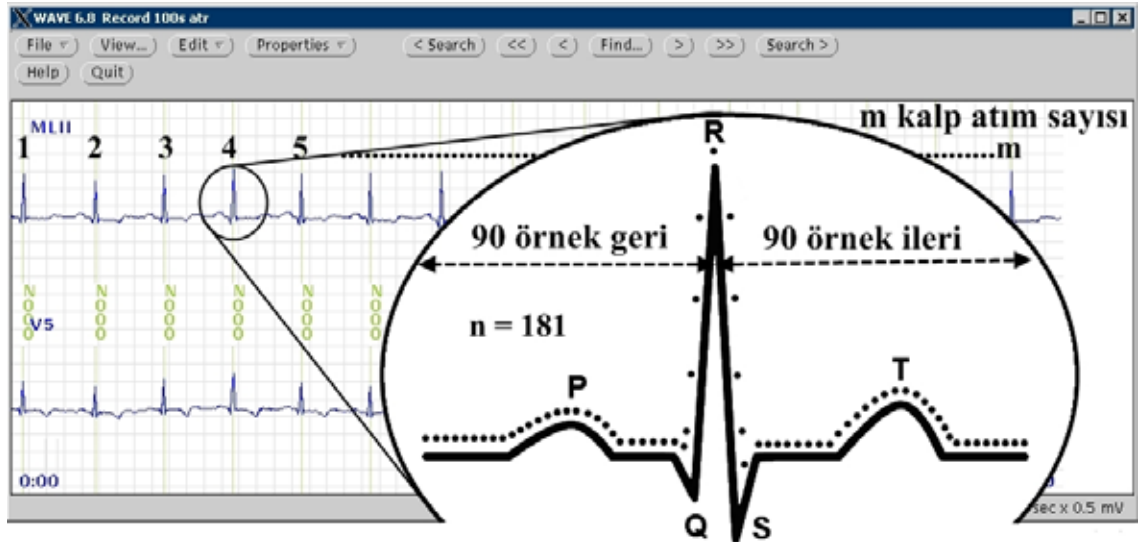
Bu tez çalışmasında, MIT-BIH EKG veritabanından alınan ve çalışmada ham veri dosyaları olarak ifade edilen 205, 208, 210 ve 213 numaralı hasta kayıtları kullanılmıştır. Bu hastalara ait EKG kayıtların seçilmesinin sebebi N, V ve F olmak üzere toplam üç sınıftan normal, PVC ve füzyon tipi atımları zengin bir şekilde bulundurmalarıdır. Ham veri seti 9.221 normal, 1.477 PVC ve 756 füzyon atımdan meydana gelmektedir. Bu kayıtların aritmi içerikleri Tablo 3.2'de AAMI sınıflandırılması altında, aritmi türlerine göre verilmiştir.

Tablo 3.2. Tez çalışmasında kullanılan ham dosyaların aritmi içerikleri.

AAMI Sınıflandırması	N	V	F
MIT-BIH Aritmi Tipleri	N	PVC	F
Ham Veri Seti*	9.221	1.477	756
205	2.571	71	11
208	1.585	992	373
210	2.423	194	10
213	2.641	220	362

*Ham veri seti 205, 208, 210 ve 213 numaralı kayıtların toplamıdır.

Ham veri seti olarak kullanılan 205, 208, 210 ve 213 numaralı kayıtlardan her bir atım, R tepeleri etrafında 90 örnek geri ve 90 örnek ileri gitmek sureti ile R tepesi örneği de dâhil toplam 181 örnek uzunluğunda vektörler şeklinde örneklenmiştir. Kalp atımları için örnekleme işlemi Şekil 3.3'te verilmiştir. Burada m kayıt içindeki atım sayısını, n ise örnek sayısını ifade etmektedir. Yarım saatlik her bir kayıt için m değeri sabit değildir. Çünkü kalp atım sayısı kişiden kişiye değişiklik gösterir [50]. Tez çalışmasında kullanılan bütün kalp atımları 181 örnekle ifade edildiği için n sabit ve değeri 181'dir.



Şekil 3.3. EKG işaret işlemede izlenen adımlar.

3.4.2. TBA Yönteminin EKG Kayıtlarına Uygulanması

TBA yöntemi, n örnek uzunluklu bir x verisini p sayıda temel bileşen ($T_1, T_2, \dots, T_p$) ile ifade etmek için kovaryans (ilişki) matrisinin özdeğer ve özvektörlerini kullanır. Burada $p \leq n$ 'dir. TBA algoritması kovaryans, standart sapma, özvektör ve özdeğer aritmetik hesaplamalarını gerektirir. Bu tez çalışmasında kullanılan EKG verilerine TBA yöntemin nasıl uygulandığı aşağıda formüle edilmiştir [16, 64, 65].

TBA yönteminin EKG kayıtlarına nasıl uygulandığını ifade etmek için MIT-BIH veritabanındaki kayıtların nasıl işlendiği aşağıda ayrıntılı olarak verilmiştir. Şekil 3.3'te verilen $n = 181$ örnek uzunluklu m adet kalp atımının her biri, $x_1 x_2 \dots x_m$ vektörlerini temsil eder. Şekil 3.3 temsili bir gösterim olup, m değeri sadece bir kayıt için değil 205, 208, 210 ve 213 numaralı toplam dört adet yarım saatlik EKG kaydının birbiri arkasına eklenmesi ile elde edilir. m adet kalp atımından oluşan X verisi Denklem 3.3'te verildiği gibi oluşturulur.

Bu tez çalışmasında üç sınıftan (N, V ve F tipi aritmiler) verilerin bulunduğu ham veri seti (205, 208, 210 ve 210 numaralı kayıtlar) için $m = 11.454$ değerindedir. Ham veri seti yani X matrisi 9.221'i N, 1.477'si V ve 756'sı ise F sınıfından olmak üzere toplam 11.454 adet 181 örnek uzunluğunda vektörden oluşur. Burada x_1 vektörü birinci kalp atımını, x_2 vektörü ikinci kalp atımını ve en nihayetinde x_m vektörü ise 11.454'inci kalp atımını temsil eder.

$$X = \begin{bmatrix} x_1 & x_2 & \dots & x_m \end{bmatrix} \quad (3.3)$$

m sayıda, 181 örnek uzunluklu x vektörü 11.454 adet kalp atımından oluşan X matrisini meydana getirmektedir. X matrisi içerisindeki her bir sütun bir kalp atımına karşılık gelmektedir. $n=181$ örnek uzunluklu x_m vektörü, Denklem 3.4 ile temsil edilir. Burada τ transpoz alma operatörünü temsil eder.

$$x_m = \begin{bmatrix} x(1)_m & x(2)_m & \dots & x(n)_m \end{bmatrix}^{\tau} \quad (3.4)$$

Transpozu alınan n örnek uzunluğundaki m adet vektörün oluşturduğu n*m boyutlu X matrisi Denklem 3.5'te verilmiştir.

$$X = \begin{bmatrix} x(1)_1 & x(1)_2 & x(1)_3 & . & . & . & x(1)_m \\ x(2)_1 & x(2)_2 & x(2)_3 & . & . & . & x(2)_m \\ x(3)_1 & x(3)_2 & x(3)_3 & . & . & . & x(3)_m \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ x(n)_1 & x(n)_2 & x(n)_3 & . & . & . & x(n)_m \end{bmatrix}_{n*m} \quad (3.5)$$

X verisine TBA yöntemi uygulamak için, verilen denklem çözümlmeleri uygun sıraya göre takip edilmelidir. Bu sıra aşağıdaki gibi işletilmelidir;

İlk adım olarak X matrisinin ortalama değerinin, $\bar{X}$, Denklem 3.6'daki gibi hesaplanması gerekir.

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n (x_i) \quad (3.6)$$

Ortalama değer vektörünün hesaplanması Denklem 3.7'de açıkça gösterilmektedir. Burada her sütun kendi içerisinde toplanır ve eleman sayısına bölünerek her sütun için bir ortalama değer bulunur. n*m boyutlu X matrisi için hesaplanan $\bar{X}$ ortalama değeri m uzunluğunda bir vektördür.

$$\bar{X} = \frac{1}{n} \left[\begin{bmatrix} x(1)_1 \\ x(2)_1 \\ . \\ + \\ . \\ x(n)_1 \end{bmatrix} \begin{bmatrix} x(1)_2 \\ x(2)_2 \\ . \\ + \\ . \\ x(n)_2 \end{bmatrix} \begin{bmatrix} x(1)_3 \\ x(2)_3 \\ . \\ + \\ . \\ x(n)_3 \end{bmatrix} . . . \begin{bmatrix} x(1)_m \\ x(2)_m \\ . \\ + \\ . \\ x(n)_m \end{bmatrix} \right]_{n*m} = [\bar{x}_1 \quad \bar{x}_2 \quad \bar{x}_3 \quad . \quad . \quad . \quad \bar{x}_m]_m \quad (3.7)$$

X verisinden ortalaması, yani $\bar{X}$ değeri çıkartılarak NX öznitelik matrisi bulunur. $NX = X - \bar{X}$, Denklem 3.8'deki gibi X matrisinden $\bar{X}$ vektörünün çıkarılması ile elde edilir.

$$NX = X - \bar{X} = \begin{bmatrix} x(1)_1 - \bar{x}_1 & x(1)_2 - \bar{x}_2 & x(1)_3 - \bar{x}_3 & . & . & . & x(1)_m - \bar{x}_m \\ x(2)_1 - \bar{x}_1 & x(2)_2 - \bar{x}_2 & x(2)_3 - \bar{x}_3 & . & . & . & x(2)_m - \bar{x}_m \\ x(3)_1 - \bar{x}_1 & x(3)_2 - \bar{x}_2 & x(3)_3 - \bar{x}_3 & . & . & . & x(3)_m - \bar{x}_m \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ x(n)_1 - \bar{x}_1 & x(n)_2 - \bar{x}_2 & x(n)_3 - \bar{x}_3 & . & . & . & x(n)_m - \bar{x}_m \end{bmatrix}_{n \times m} \quad (3.8)$$

NX öznitelik matrisinin her bir sütunu transpozu ile çarpılarak kovaryans matrisi bulunur. Kovaryans matrisinin hesabı Denklem 3.9'da verilmiştir. $(x_i - \bar{x}_i)$ elemanının transpozu $(x_i - \bar{x}_i)^T$ ile ifade edilir.

$$C_x = \frac{1}{m} \sum_{i=1}^m \{(x_i - \bar{x}_i)(x_i - \bar{x}_i)^T\} = \frac{1}{m} \sum_{i=1}^m \{(\Delta x_i)(\Delta x_i)^T\} \quad (3.9)$$

Kovaryans matrisinin her bir sütunu Denklem 3.10'daki gibi ifade edilir.

$$\Delta x_i = (x_i - \bar{x}_i) = \begin{bmatrix} x(1)_i \\ x(2)_i \\ . \\ . \\ x(n)_i \end{bmatrix} - [\bar{x}_i] = \begin{bmatrix} x(1)_i - \bar{x}_i \\ x(2)_i - \bar{x}_i \\ . \\ . \\ x(n)_i - \bar{x}_i \end{bmatrix} = \begin{bmatrix} \Delta x(1)_i \\ \Delta x(2)_i \\ . \\ . \\ \Delta x(n)_i \end{bmatrix} \quad (3.10)$$

Bu değerin transpozu Denklem 3.11'de verilmiştir.

$$\Delta x_i^T = (x_i - \bar{x}_i)^T = [\Delta x(1)_i \quad \Delta x(2)_i \quad . \quad . \quad \Delta x(n)_i] \quad (3.11)$$

Kovaryans matrisinin açık ifadesi Denklem 3.12’de verilmiştir.

$$C_x = \frac{1}{m} \begin{bmatrix} \Delta x(1)_1 & \Delta x(1)_2 & . & . & \Delta x(1)_m \\ \Delta x(2)_1 & \Delta x(2)_2 & . & . & \Delta x(2)_m \\ . & . & . & . & . \\ . & . & . & . & . \\ \Delta x(n)_1 & \Delta x(n)_2 & . & . & \Delta x(n)_m \end{bmatrix}_{n*m} \begin{bmatrix} \Delta x(1)_1 & \Delta x(2)_1 & . & . & \Delta x(n)_1 \\ \Delta x(1)_2 & \Delta x(2)_2 & . & . & \Delta x(n)_2 \\ . & . & . & . & . \\ . & . & . & . & . \\ \Delta x(1)_m & \Delta x(2)_m & . & . & \Delta x(n)_m \end{bmatrix}_{m*n} \quad (3.12)$$

$n*m$ ve $m*n$ boyutlu bu iki matrisin çarpımı sonucunda $n*n$ boyutlu bir matris elde edilir. $n*n$ boyutlu kovaryans matrisi, elemanları ile birlikte Denklem 3.13’te verilmiştir.

$$C_x = \begin{bmatrix} Cx(1)_1 & Cx(1)_2 & . & . & Cx(1)_n \\ Cx(2)_1 & Cx(2)_2 & . & . & Cx(2)_n \\ . & . & . & . & . \\ . & . & . & . & . \\ Cx(n)_1 & Cx(n)_2 & . & . & Cx(n)_n \end{bmatrix}_{n*n} \quad (3.13)$$

Denklem 3.14’ten simetrik kovaryans matrisinin özvektörleri P_i ve özdeğerleri λ_i bulunarak, kovaryans matrisinin ortagonal temeli bulunabilir.

$$C_x P_i = \lambda_i P_i \quad i=1, 2, \dots, 8 \quad (3.14)$$

λ_i özdeğerleri Denklem 3.15 ile hesaplanabilir. Burada I birim matristir. Bu denklem çözümlenerek kovaryans matrisi ile birim matrisinin farkının determinantını sıfıra eşitleyen değerler bulunur. Elde edilen bu değerler, λ_i özdeğerlerini vermektedir.

$$\det(C_x - \lambda I) = 0 \quad (3.15)$$

Birinci temel bileşen P_1 varyansı en çok açıklayan en büyük özdeğere λ_1 karşılık gelen özvektördür. Temel bileşenlerin, özdeğerler ile olan ilişkisi Denklem 3.16'da verilmiştir.

$$\text{Var}(P_1) = \lambda_1 \quad (3.16)$$

Bu tez çalışmasında, YSA boyutunu küçültmek amacı ile tek kanal EKG işaretine TBA öznitelik çıkarım yöntemi uygulanmıştır. Ham veri setindeki 181 örnek uzunluklu kalp atımları, TBA yöntemi kullanılarak daha az sayıda veri ile yani TB'ler ile ifade edilmiştir. Beşinci bölümde TBA yönteminin EKG işaretlerine nasıl uygulandığı gösterilmiştir. Temel bileşenler ile veri setindeki değişim görselleştirilmiş ve veri boyutunun ne kadar etkili bir şekilde azaltıldığı gösterilmiştir.

4. BÖLÜM

SINIFLANDIRMA YÖNTEMİ OLARAK YSA

EKG işaretleri literatürde farklı yönleri ile yorumlanmış ve çok sayıda çalışma yapılmıştır. Bu tez çalışmasında EKG işaretlerindeki bozuk ritimler yani aritmiler incelenmiştir. 3. Bölüm’de EKG işaret işleme alanları ve öznitelik çıkarım yöntemleri verilmiştir. Öznitelik çıkarım yöntemleri genellikle uygun bir sınıflandırma algoritması ile birlikte kullanılırlar. Aritmi sınıflandırma uygulamalarında kullanılan sınıflandırma yöntemleri ve literatür özeti aşağıdaki gibi verilmiştir.

1961 yılında Cady ve arkadaşları tarafından QRS komplekslerinin matematiksel bir teknik olarak fourier analizi ile sınıflandırılması çalışması yapılmıştır [66]. 1964 yılında aritmi tanımda adaptif filtre kullanımı konusundaki ilk çalışmalardan biri Yasui ve arkadaşları tarafından yapılmıştır [67]. Son yirmi yıl içerisinde EKG işaretlerinin analizinde yeni birçok teknik önerildi. Bunlara örnek olarak; YSA [14-18, 34, 68-71], GA [8, 72], filterbanks [73], HMM [23], doğrusal ayırtaç fonksiyonu [24], adaptif filtre [25, 74], destek vektör makinesi [26, 75], doğrusal kestirim [27, 76], bayes tahmini [10], çoğu doğrusal olmayan dönüşüm tabanlı sezgisel [77, 78] ve istatistiksel [5] yöntemler verilebilir.

YSA’lar sahip oldukları öğrenme, genelleme ve altıncı duyu olarak ifade edilebilecek tahmin yapabilme özelliklerinden dolayı birçok mühendislik alanında, hatta günümüzde sosyal alanlarda dahi geniş uygulama alanı bulmuştur. Özellikle doğrusal olmayan sistemleri modellerken matematiksel bir formülizasyona ihtiyaç duymaması YSA mimarisinin basit ve hızlı olmasını sağlar. Güçlü tahminsel yeteneğinden dolayı Medikal Tanı Sistemlerinde (MTS) en çok kullanılan yöntem YSA’lardır. Birçok araştırmacı YSA’nın performansını diğer metotlar ile karşılaştırmış ve YSA’nın diğerlerinden daha iyi sonuçlar ürettiğini göstermişlerdir [33].

TBA ve YSA, genelleme ve tahmin uygulamalarında en sık kullanılan öznitelik çıkarım ve sınıflandırma yöntemleridir. TBA ile elde edilen az sayıda veri ile eğitilmiş bir YSA, yüksek doğruluklarda sınıflandırma yapabilmektedir. YSA'nın sahip olduğu basit ve paralel mimari, FPGA gerçeklemeleri için de oldukça uygundur.

4.1. YSA'nın Kullanım Alanları ve Tarihsel Gelişimi

YSA insan beyninin yapısı ve fonksiyonlarını temel alan bir bilgi işleme sistemi ve birçok alanda problem çözümlerinde kullanılan popüler bir araçtır. İnsan beyninde yaklaşık 10 milyar nöron ve 100 trilyon bağlantı bulunmaktadır [79]. Bilgiler dağıtık olarak nöronların gövdelerinde bulunurlar ve bu dağıtık veriler sistem içerisinde paralel olarak işletilmektedir. İnsan beynindeki biyolojik yapının öğrenmesinden esinlenilerek tasarlan YSA, geleneksel yöntemlerin başarısız olduğu birçok alanda problem çözme uygulamalarına başarılı bir şekilde uygulanmış bir hesaplama yöntemidir. Ses tanıma, görüntü işleme, desen tanıma ve sınıflandırma, finans eğilimi tahminleri, karma yapıların üretilmesi, süreç modelleme, kontrol, biyomedikal gibi alanlar bu uygulamalara örnek verilebilir [80-83].

YSA ile ilgili çalışmalar biyolojik nöronun tanımlanması ile başlar. 1943 yılında nörobiyolog McCulloch ve matematikçi Pitts “Sinir Aktivitesindeki Düşüncelere Ait Mantıksal Bir Hesap” isimli çalışmalarıyla, bir biyolojik nöronun temel fonksiyonlarının, eşik seviyeleri ile tanımlanabileceğini önermişlerdir [84]. 1949 yılında Donald Hebb “Davranış Organizasyonu” adlı kitabı ile beynin öğrenme sistemini hücresele seviyede tanımlamıştır [85]. Biyolojik bir nöronun temel çalışma sistemi ve ardından bilinen ilk tek katmanlı YSA olarak tanımlanan, basit mantıksal yorumlar yapan iki katmanlı bir hesaplama ünitesi olan perceptron, 1958 yılında psikolog Frank Rosenblatt tarafından bulunmuştur [86].

1959 yılında Widrow basit nöron benzeri birimlerden oluşan Adaline (Adaptive Linear Neuron) ve Madaline (Multiple Adaline) diye isimlendirilen, ilk defa hava tahmini ve ses tanıma uygulamalarında kullanılan bir doğrusal eleman geliştirmiştir [87]. 1960 yılında Widrow ve Hoff bu basit nöron kuramını kullanarak öğrenebilen ilk adaptif sistem üzerine çalışmışlar ve delta kuramı olarak bilinen gerçek çıkış ile istenilen çıkış arasındaki farka eşit bir hata terimi kullanarak, bağ ağırlıklarının değiştiği bir öğrenme kuralını ortaya koymuşlardır [88].

1969 yılında Minsky ve Papert basit perceptron modelinin doğrusal problemlere uygulanabildiğini ancak bu model ile klasik XOR probleminin çözülmesinin imkânsız olduğunu göstermiş ve YSA ile öğrenme ve hesaplamada aşılması güç engeller olduğunu iddia etmişlerdir. Bu iddiaları YSA ile ilgili yapılan çalışmaları durma noktasına getirmiştir [89].

Modern YSA dönemi 1982’de bir fizikçi olan Hopfield tarafından yayınlanan “Neural Network and Physical System” adlı çalışma ile başlamıştır [90]. Bu çalışmada Hopfield, moleküler biyolojinin karmaşık izahlarından farklı olarak, nöronların karşılıklı etkileşimine dayanan bir hesaplama modeli önermiştir. Daha sonra 1987 yılında geri beslemeli YSA modelini ortaya koymuş ve bunu gerçek problemlere uygulamıştır [91].

1984 yılında Kohonen kendi kendini düzenleyen nitelik haritası (SOM, Self-Organizing Map) kuramını sunmuştur [92]. 1986 yılında Gossberg Uyarlanabilir Rezonans Teorisi (ART, Adaptive Resonance Theory) isimli bir YSA yapısı geliştirmiştir [93]. 1986 yılında Rumelhart ve McClelland “Parallel Distributed Processing” adlı kitap ile İleri Beslemeli (FW, Feed Forward) ağlarda yeni bir öğrenme modeli olarak, hatanın Geriye Yayılımı (BP, Back Propagation) algoritmasını yayınlamışlardır. Minsky ve Papert tarafından iddia edilen aksaklıkların aşılabileceğini bu çalışmaları ile ispat etmişlerdir [94]. BP algoritması ile daha zor desenleri sınıflandırabilecek yeteneğe sahip eğitilebilen bir YSA oluşturmak için, giriş ile çıkış katmanları arasına bir gizli katman eklenmiştir. BP algoritması bugün kullanılan en popüler yöntemdir [95].

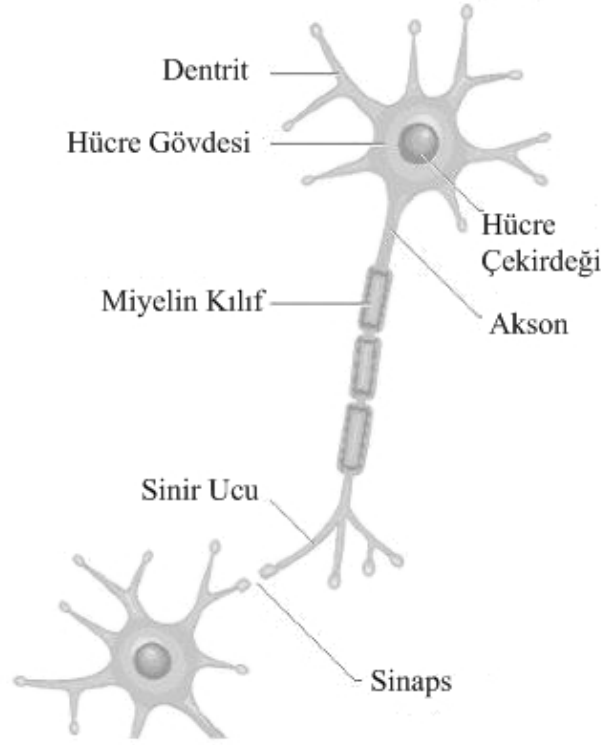
4.2. Biyolojik Sinir Ağı

1943 yılında McCulloch ve Pitts’in biyolojik nöronu tanımlanması ile başlayan çalışmalar [84], konunun farklı biçimlerde yorumlanması ile devam etmiştir. 1952 yılında Hodgkin ve Huxley mürekkep balığının dev aksonunun elektriksel modelini çıkardılar ve bu çalışmaları ile 1963 yılında Nobel ödülünü aldılar [96]. Biyolojik hücrelerin davranışlarının matematiksel yorumlarının çıkarılması 1950’lerden beri devam eden bir çalışma alanıdır. Nöron hücreleri araştırmacıların en çok ilgisini çeken yapılardır. Nöronlar tek başlarına değil bir ağın bileşenleri olarak incelendiğinde anlamlıdırlar. Gerek insan beyinde sayıları 10 milyarı bulan nöronlar gerekse diğer canlılarda bulunan daha az sayıdaki nöronlar, bir ağın parçası olarak canlının karar mekanizmasını işletirler. Örneğin Kerevit denilen canlıda bir beyin bulunmaz fakat 11

adet nörondan oluşan küçük bir sinir ağına sahiptir ve bu küçük sinir sistemi canlının temel fonksiyonlarını yönetir. Canlıların sinir sistemlerinin incelenmesi için kullanılan birçok yöntem olmasına karşın, bu çalışmalar “In-Vivo” ve “In-Vitro” olmak üzere iki ana başlık altında incelenebilir. In-Vivo çalışmalarında keskin elektrot dizilerinden oluşan bir kask, canlının beynine gömülür ve beynin çeşitli bölgelerinden elektriksel aktiviteler okunur ya da beynin çeşitli bölgeleri dışarıdan verilen sinyaller ile uyarılır [97]. Canlıların beyninden alınan işaretlerden dış etkilere ürettikleri tepkiler incelendiği gibi, beyne dışarıdan uygulanan elektriksel uyarıların canlıların davranışlarına olan etkileri de incelenmektedir. In-Vitro çalışmalarda ise hayvandan alınan bir parça doku veya bir miktar nöron hücresi bir dış ortamda çalıştırılır. Bu çalışmaların en ilginç olanlarından biri hayvandan alınan sinir hücreleri ile elektriksel olarak iletişim kurmak üzerindedir. Bu hücreler ile iletişim kurma yollarından biri, bir cama yapıştırılmış metal elektrot dizileri üzerine canlı nöron hücrelerinin yerleştirilmesi yöntemidir. Bu yolla metal elektrotlar nöron hücreleri ile elektriksel olarak bağlanmış olur ve nöronlara elektriksel olarak etki edilebilir veya davranışları dış ortamlardan ölçülebilir. Bu yöntem sayesinde nöron hücrelerinin matematiksel modelleri, elektriksel uyarılara verdikleri cevaplar incelenerek ortaya konulabilir. Ayrıca canlı sinir hücrelerinin silikon transistörler ile birlikte kullanılmasına da imkân tanıyan bu uygulamalar, sinir ağlarının davranışları hakkında çok önemli bilgilerin edinilmesi adına oldukça ilgi çekicidir [98].

Biyolojik sinir ağlarındaki nöron bağlantıları yapay sinir ağlarında olduklarından çok daha karmaşıktır. Şekil 4.1’de temel bir biyolojik nöron hücresi gösterilmiştir. Bir nöron hücresi hücre çekirdeğini saran hücre gövdesi, dentrit ve akson bileşenlerinden meydana gelir. Hücre gövdesi nöronun kalbidir ve dentritler vasıtası ile diğer nöronlardan toplanan elektriksel sinyallere üretilecek tepkiyi belirler. Hücre gövdesinde üretilen cevap, çok sayıda dallara ayrılan uzun akson iletim yolları üzerinden diğer hücrelere taşınır. Aksonlardan dallanan çok sayıdaki iletim yolu sinir ucu olarak isimlendirilir ve diğer nöronların bilgi alıcı uçları olan dentritler ile sinapslar üzerinden bağlanarak, verinin diğer nöronlara veya terminal organlara iletilmesini sağlarlar. Bir akson birden fazla dentrit ile sinapslar üzerinden bağlanabilir. Sinapslar nöronlar arasındaki elektrokimyasal bağlantıyı sağlarlar. Bir elektriksel iletinin sinapslar üzerinden diğer nöronlara aktarılması için elektriksel işaretin, sinaptik bağlantı noktasına özgü belli bir

eşiğin üzerinde olması gerekir. Sinapslardaki genliğin, eşiği geçerek iletilmesi durumuna “nöronun ateşlenmesi” denir. Sinapslara ulaşan elektriksel işaret eğer arttırılarak dentritlere iletiliyorsa uyarıcı, azaltılıyor ise önleyici olarak isimlendirilirler [99].



Şekil 4.1. Biyolojik nöron [99].

Nöronlar merkezi sinir sisteminin en temel bileşenidir. Biyolojik sinir ağı, vücudun çeşitli bölgelerinde oluşan uyarıları merkezi sinir sistemine taşıyan alıcı nöronlar ve merkezi sinir ağında yorumlanmış uyarıları ilgili cevap bölgelerine taşıyarak, gerekli tepkilerin üretilmesini sağlayan tepki nöronlarından meydana gelmektedir. Alıcı nöronlar reseptör, tepki nöronları ise efektör olarak da isimlendirilirler. Biyolojik sinir ağı, organizmadan reseptörler ile toplanan bilgiyi sürekli olarak yorumlar ve gerekli cevapları reseptörler vasıtası ile organizmanın ilgili noktalarında uygun tepkilere dönüştürür. Biyolojik nöronların hızları yapay nöronlardan daha düşük olmasına karşın merkezi sinir sistemi ve beyin arasındaki mükemmel veri organizasyonu sayesinde, verilerin değerlendirilmesi yapay sinir ağlarında olduklarından daha kısa sürelerde oluşmaktadır [79]. Şekil 4.2’de merkezi sinir sistemindeki veri akışı gösterilmektedir.



Şekil 4.2. Merkezi sinir sisteminde veri akışı.

4.3. Yapay Sinir Ağlarının Özellikleri

Biyolojik sinir ağının büyüleyici gücü araştırmacıların YSA geliştirme çalışmalarında büyük gayretler göstermelerine sebep olmaktadır. YSA insan beynindeki yapıyı temel alan bir modeldir ve insan beynindeki dağıtılmış paralel işlemi taklit etmek amacıyla, biyolojik nöronları temsil eden işlemci elemanların değişik şekillerde birbirleri ile bağlanması yoluyla işletilir. Bağlantılar genelde katmanlar ile düzenlenir. YSA ağırlıklar ile çarpılmış girişlerin işlendiği, birbirinden bağımsız bir grup işlem biriminden meydana gelir. Bu birimler girişlerden ve diğer birimlerin çıkışlarından gelen ağırlıklandırılmış girişler ile birbirlerine bağlanırlar. YSA'nın tahmin gücü, bu bağımsız hesaplama birimleri ve karmaşık bağlantılarından gelmektedir.

YSA'lar eğitildikten sonra daha önce eğitim verisi olarak uygulanmamış girişlere dahi uygun çıkışlar üretebilmektedirler. Bu durum YSA'ların genelleme özelliğini ifade eder. YSA'lar gerçek veriler ile modellendiklerinde ürettikleri sonuçlar gerçekçidir [100]. Girişler ile çıkışlar arasındaki ilişkiyi en iyi şekilde tanımlayacak şekilde ağırlıkların oluşturulması işlemi, YSA'nın öğrenme özelliği olarak isimlendirilir. Bu işlem farklı eğitim algoritmaları kullanılarak yapılabilir. YSA'ların bir diğer özelliği de doğrusal olmamalarıdır. Bu özellik işlemci eleman içerisindeki aktivasyon fonksiyonu ile kazandırılır. İşlemci eleman içerisinde kullanılan aktivasyon fonksiyonu doğrusal değilse YSA'nın cevabı da doğrusal olmayacaktır. Sabit bir YSA modeli, farklı problemleri çözmek için ilgili probleme ait veriler ile eğitilerek kullanılabilir. Bu durum YSA'ların uyarlanabilirlik özelliğidir. YSA'ların hataya karşı bağımsızlıkları ise bir diğer özellikleridir. Hata bağımsızlığı ağın boyutu ile doğru orantılıdır. Yani ağın boyutu ne kadar büyük olursa, içerideki herhangi bir işlemci elemanın çıkışa olan katkısı o denli küçük olacak ve ağın hataya karşı bağımsızlığı da o denli yüksek olacaktır.

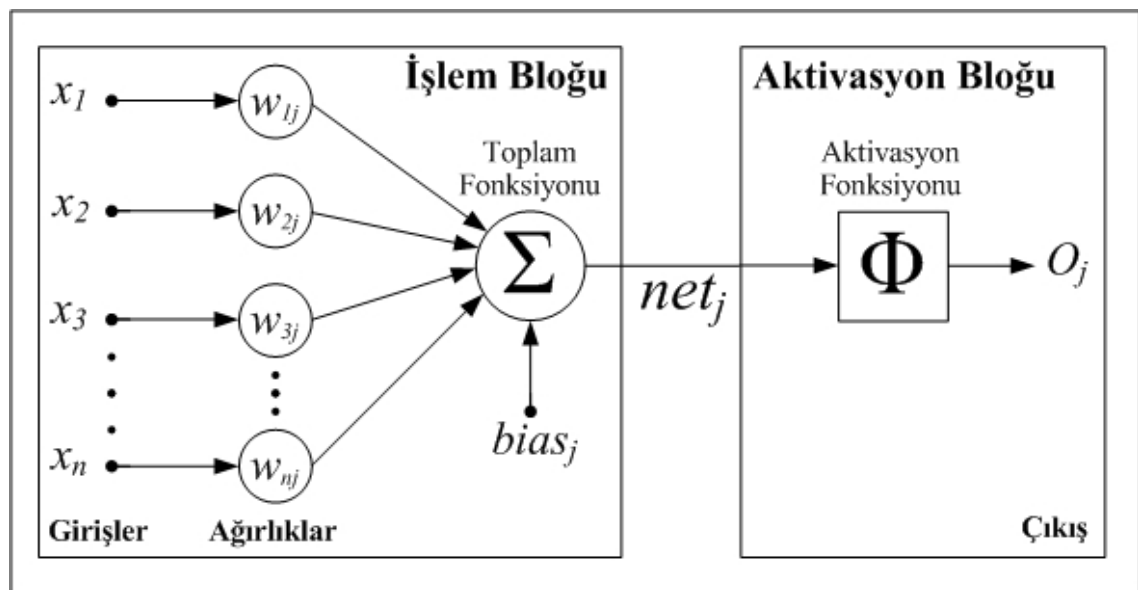
YSA'nın dağıtık işlem özelliği ve buna bağlı cevap hızı, geleneksel mikroişlemcilerle değil ancak paralel donanımlar ile elde edilebilir.

YSA tanımlanırken, önemli üç karakteristik bileşeninin tanımlanması gerekmektedir.

- i) Yapay Nöron Modeli,
- ii) Ağ Yapısı,
- iii) Eğitim Algoritması.

4.3.1. Yapay Nöron Modeli

Yapay nöron işlemci eleman olarak isimlendirilir ve Şekil 4.3'te gösterildiği gibi işlem ve aktivasyon bloklarından meydana gelir. İşlemci eleman, girişleri ile çıkışı arasındaki ilişkiyi, sahip olduğu aktivasyon fonksiyonuna göre belirler. Bu bakımdan işlemci elemanın davranışında aktivasyon fonksiyonu çok önemli bir rol oynar. Aktivasyon fonksiyonları temel anlamda üç gruba ayrılabilir. Bunlar doğrusal veya rampa (yarı doğrusal), eşik ve doğrusal olmayan aktivasyon fonksiyonlarıdır. Bu aktivasyon fonksiyonları içerisinde biyolojik nöronun davranışına en yakın cevaplar, doğrusal olmayan aktivasyon fonksiyonları ile sağlanır. Doğrusal olmayan sigmoid aktivasyon fonksiyonları biyolojik nöronun cevabına en yakın çıkışları üretir ve bu yumuşak geçişli çıkış karakteristiği, doğrusal olmayan sigmoid fonksiyonunu uygulamalarda en çok kullanılan aktivasyon fonksiyonu yapar [101].



Şekil 4.3. İşlemci eleman.

Yapay sinir ağlarında giriş işlemci elemanları, X_i giriş sinyali gizli katmanlara taşımaktan sorumludur. Gizli katmandaki j işlemci elemanı, X_{ij} girişlerini W_{ij} ağırlık değerleri ile çarparak, işlem bloğu içerisinde gösterilen bir toplama fonksiyonuna uygular. Toplam sinaptik çıkış, ağırlıklandırılmış girişlerin toplamı ile $bias_j$ değerinin toplanması sonucu elde edilir ve net_j ile ifade edilir.

$$net_j = bias_j + \sum_{i=1}^n w_{ij}x_i \quad (4.1)$$

Ağırlıklandırılmış giriş verisi ile eşik toplamı, net_j , bir aktivasyon fonksiyonundan geçirilerek işlemci elemanın çıkışı, O_j , elde edilir.

$$O_j = \Phi(net_j) \quad (4.2)$$

Denklem 4.2'deki Φ , aktivasyon fonksiyonunu ifade eder. Toplam sinaptik çıkış, net_j , Φ aktivasyon fonksiyonu ile O_j değerine dönüştürülür. Biyolojik nöron ile yapay nöron arasındaki ilişki Tablo 4.1'de verilmiştir.

Tablo 4.1. Biyolojik nöron ile yapay nöron arasındaki ilişki.

Biyolojik Nöron	Yapay Nöron
• Nöron • Dendrit • Hücre Gövdesi • Aksonlar • Sinapslar	• İşlemci Eleman • Toplama Fonksiyonu • Aktivasyon Fonksiyonu • Yapay Nöron Çıkışları • Ağırlıklar

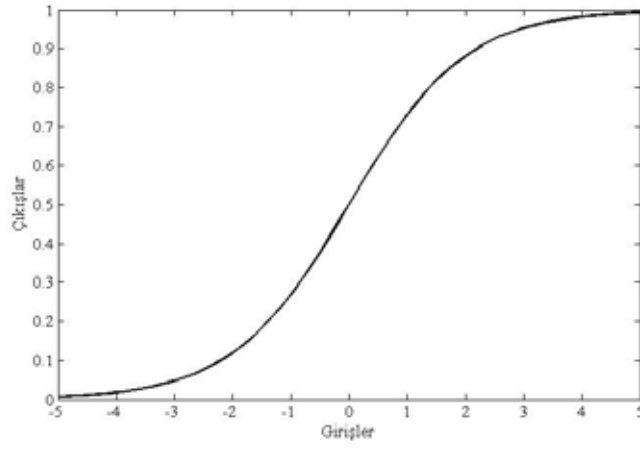
4.3.1.1. Aktivasyon Fonksiyonları

İşlemci elemanın davranışını belirleyen en önemli birim aktivasyon fonksiyonudur. En çok kullanılan aktivasyon fonksiyonları logaritmik sigmoid (logsig), tanjant sigmoid (tansig), keskin sınırlandırıcı (hardlim), üçgen (tribas), radyal (radbas), doğrusal saturasyon (satlins) ve doğrusal (purelin) fonksiyon olarak verilebilir. Bunlardan logsig ve purelin aşağıda izah edilmiştir. Doğrusal olmayan aktivasyon fonksiyonlarının kullanımı, YSA'nın daha karmaşık problemlere uygulanmasını mümkün kılmaktadır. Aktivasyon fonksiyonlarına öğrenme eğrisi de denilmektedir. Öğrenme eğrileri türevlenebilir sürekli fonksiyonlardır. Doğrusal olan veya olmayan problemlerin çözümlerinde kullanılmak üzere bu eğrilerin herhangi biri seçilebilir [102].

4.3.1.1.1. Logaritmik Sigmoid Aktivasyon Fonksiyonu

Logaritmik sigmoid en çok tercih edilen aktivasyon fonksiyonudur. Matlab komut satırından `logsig` komutu ile işletilir. `logsig` fonksiyonunun matematiksel modeli Denklem 4.3'te, fonksiyonun davranış cevabı ise Şekil 4.4'te görülmektedir.

$$O_j = \frac{1}{1 + e^{-net_j}} \quad (4.3)$$

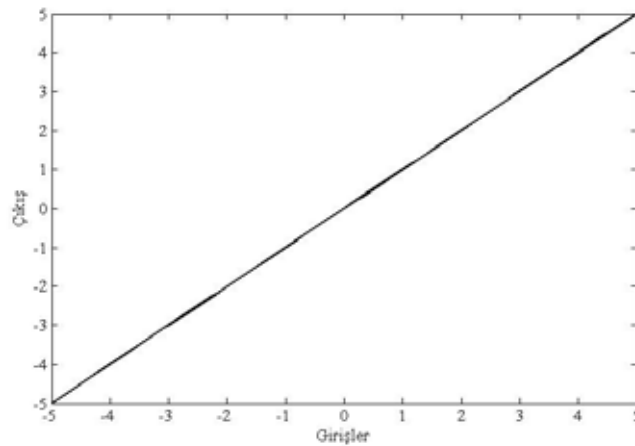


Şekil 4.4. Logaritmik sigmoid aktivasyon fonksiyonu.

4.3.1.1.2. Doğrusal Aktivasyon Fonksiyonu

Doğrusal, purelin, aktivasyon fonksiyonunun matematiksel modeli Denklem 4.4'te, fonksiyonun çıkış cevabı ise Şekil 4.5'te verilmiştir.

$$purelin = net_j \quad (4.4)$$



Şekil 4.5. Purelin aktivasyon fonksiyonu.

4.4. Biyolojik ve Yapay Sinir Ağlarının Kabiliyetleri

Biyolojik sinir sisteminde yaklaşık olarak 10 milyar nöron ve 100 trilyon sinaptik bağlantının olduğu daha önce ifade edilmişti. Bunun anlamı her bir nöron ortalama 10.000 sinaps ile bağlantı kurmasıdır. Bu değerler bireylerin nöro anatomik yapısına göre büyük farklılıklar göstermektedir. Bireylerdeki algı, tepki ve zekâ farklılıkları, beyindeki nöro anatomik farklılıklardan ileri gelmektedir.

Biyolojik sinir ağı duygusal ve görsel işlemlerde bilgisayarlardan daha hızlı cevaplar üretirler. Farklı uyaranlar için beynin farklı bölgelerinde enerji akışları görülmektedir. Bu, biyolojik sinir ağının dağıtılmış paralel mimarisinden ileri gelmektedir. Farklı görevleri icra etmek amacı ile beynin farklı bölgelerinde tepkiler oluşturulur.

Tek işlemcili geleneksel bilgisayarların birim zamanda gerçekleştirdiği tek işlem mantığının aksine, biyolojik sinir ağında nöronlardan oluşan yüksek işlem güçlü bir paralel hesaplama mimarisi bulunmaktadır. Buna rağmen biyolojik sinir ağındaki her bir nöronun birim zamandaki işlem hızı, güçlü merkezi işlemcilerin (CPU, Central Processing Unit) sahip olduğu hızlardan çok daha düşüktür. Birim zamanda üretilen cevap, işlem/saniye yani i/sn olarak ifade edilecek olursa, bu değer biyolojik nöronlar için 100 i/sn iken CPU'lar için saniyede 100 milyonlarca işlemdir. CPU'dan daha yavaş bir donanıma sahip olmasına karşın aşağıda maddeler halinde verilen avantajları, beyni çok daha üstün kabiliyetli bir hesaplama mimarisi haline getirmektedir [79].

- Bilgisayar programları veya elektronik donanımlar, işlemsel bir birimlerinin bulunmaması veya bozulması durumunda işlevselliğini yitirirler. Buna karşın beyinde oluşan bölgesel hasarlar büyük bir ustalıkla tolere edilir ve işlevselliğini sürdürür.
- Beyin tecrübeler ile öğrenmeye devam eder ve sürekli olarak kendini yeniden organize eder.
- Beyin, daha önce hasar gören bölge tarafından yürütülen işlevselliği, sağlıklı bölgenin öğrenmesi sağlayarak devredebilir ve işlevsellik bu yolla devam ettirilebilir.

- Beyin yüksek seviyedeki paralel işlem birimlerini çok etkin bir biçimde işletir. Örneğin karmaşık görsel algı beyinde 10 adımda yani 100 msn gibi bir sürede tamamlanır.
- Tam olarak nasıl çalıştığı bilinmemekle birlikte, beyin insan zekâsı ve öz-farkındalığı destekler.

İnsan beyni tarafından kontrol edilen duygusal tepkiler ve sahip olduğu zekâ henüz bilgisayarlara aktarılamamış olsa dahi araştırmacılar, bilim kurgu yazarları ve okuyucu/izleyici kitlelerinin yıllardır ilgisini çeken bir alandır.

Bir yapay zekâ disiplini olan YSA, bilgisayarların kabiliyetini beynin işlevselliğine yaklaştırmak amacıyla, beynin büyük ölçüde basitleştirilmiş bilgi işleme sistemini taklit eder. Tablo 4.2’de beyin ile bilgisayar arasındaki performans ilişkisi yedi başlık altında verilmiştir.

Tablo 4.2. Beyin ile bilgisayar arasındaki performans ilişkisi [79].

	İşlemsel Eleman	Eleman Büyüklüğü	Enerji Watt	İşlem Hızı	Hesaplama Türü	Öğrenme	Zekâ
	10^{14} Sinaps	10^{-6} m	30W	100Hz	Paralel Dağıtık	Sürekli	
	10^8 Transistör	10^{-6} m	30W CPU	1GHz	Seri Sıralı	Sınırlı	

4.4.1. YSA’ların Sınıflandırılması

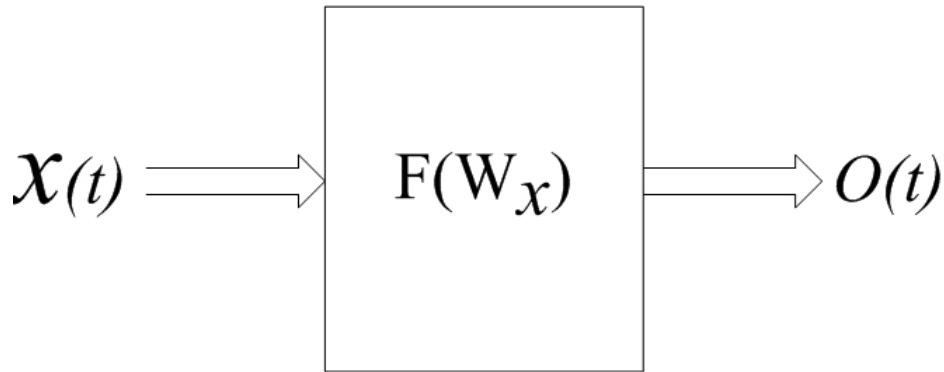
Yapay Sinir Ağları ağ yapıları ve öğrenme yöntemleri olarak iki grupta incelenebilir. Literatürde değişik katman sayısı, ağ bağlantı yapısı ve öğrenme yöntemleri bulunmaktadır. Bu yöntemlerin kısa bir özeti aşağıda verilmiştir.

4.4.1.1. YSA’ların Ağ Yapılarına Göre Sınıflandırılması

Yapay sinir ağları temel anlamda, ileri beslemeli (FeedforWard) ve geri beslemeli (Recurrent) ağlar olmak üzere iki gruba ayrılabilirler [103].

4.4.1.1.1. İleri Beslemeli Yapay Sinir Ağları

İleri beslemeli yapay sinir ağlarında bilgi işareti, işlemci elemanlar boyunca giriş katmanını, gizli katman/lar ve çıkış katmanını boyunca ileri yönlü olarak aktarılacak şekilde ağ çıkışından sonuç cevabını üretirler. Giriş katmanını dışarıdan aldığı bilgileri hiçbir değişiklik yapmadan, olduğu gibi gizli katmanlardaki işlemci elemanlara uygular. Bilgi bu katmanlar boyunca sırası ile işlenir, her bir katmanın çıkışı kendinden sonra gelen katmana giriş olarak uygulanır, herhangi bir geri besleme veya katman atlaması yoktur. İleri beslemeli YSA'lar en basit ağ yapısına sahip ve hesaplama karmaşıklığı en düşük olan ağ yapılarındandır. Bu ağ yapılarına örnek olarak Tek Katmanlı Perseptron (SLP, Single Layer Perceptron), Çok Katmanlı Perseptron (MLP, Multi Layer Perceptron), Adaline ve LVQ (Learning Vector Quantization) yapıları örnek verilebilir [104-106]. İleri beslemeli ağ yapısı Şekil 4.6'da gösterilmiştir.

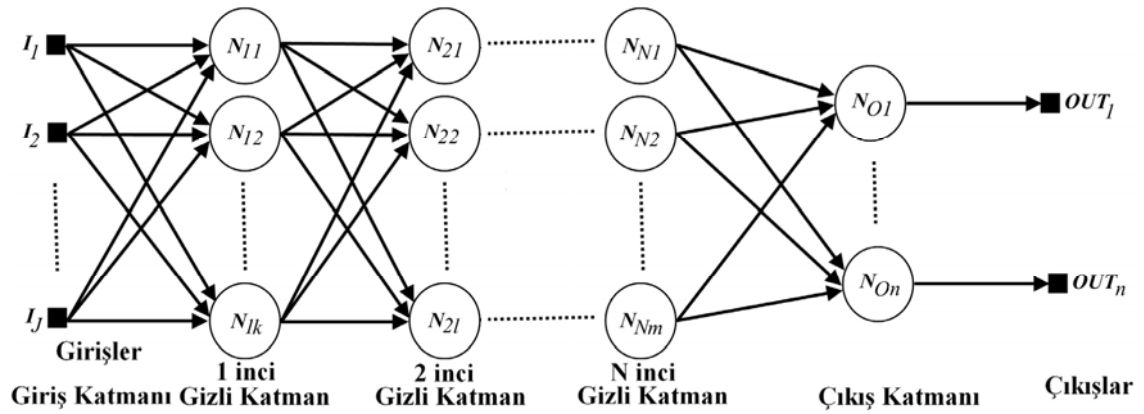


Şekil 4.6. İleri beslemeli YSA.

4.4.1.1.1.1. Çok Katmanlı Perseptron

Değişik tipte birçok ileri beslemeli YSA modeli olmasına rağmen Çok Katmanlı Perseptron (MLP) örüntü sınıflandırma uygulamalarda en çok tercih edilen mimarilerden biridir [105, 106]. MLP veri işlemede etkili ve eğitilmesi kısa sürmektedir. MLP ağlar bir giriş katmanını, bir veya daha fazla gizli katman ve bir çıkış katmanından meydana gelmektedir. Giriş katmanını bir tampon gibi davranır ve uygulanan girişleri ara katmandaki işlemci elemanlara ulaştırır. Gizli katman bir veya daha fazla sayıda katmandan oluşur ve sırasıyla birinci, ikinci, üçüncü gizli katman şeklinde isimlendirilir. Giriş katmanını veriler üzerinde herhangi bir işlem yapmadan, verileri orijinal halleri ile gizli katmana taşınmaktan sorumludur ve yeni bir tampon

olarak görev yaparlar. Fakat gizli katmanlar ve çıkış katmanı, işlemlerin yapıldığı katmanlardır. MLP ağının basitleştirilmiş bir görüntüsü Şekil 4.7’de verilmiştir. Katman sayısı ve her bir katmandaki işlemci eleman sayısı uygulamanın türüne göre değişiklik gösterebilir. Bu sebepten katman sayısı N olarak verilmiştir. Giriş, gizli katmanlar ve çıkış katmanındaki işlemci eleman sayıları ise j , k , l , m ve n şeklinde değişik harflerle temsil edilmiştir. Basit yapısı ve yüksek sınıflandırma performansı yüzünden bu tez çalışmasında MLP ağı kullanılmıştır. MLP ağlarının basit yapısı, işlemsel karmaşıklığını azaltır ve bu sayede FPGA devre gerçeklemeleri için avantajlı hale gelirler.



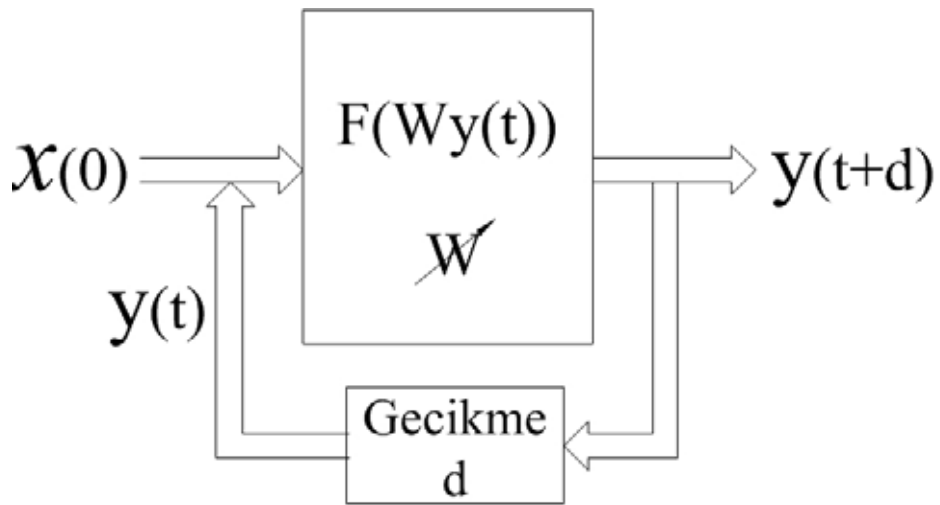
Şekil 4.7. MLP ağ yapısı.

Giriş katmanındaki I indeksli giriş verileri, ağırlık değerleri ile çarpılarak birinci gizli katmandaki işlemci elemanlara verilir. Birinci gizli katmandaki işlemci elemanlar, ağırlıklandırılmış giriş verileri ile her bir işlemci eleman için tanımlı bias değerlerini toplar ve bu sinaptik toplamı uygun bir aktivasyon fonksiyonundan geçirerek birinci gizli katmanın çıkışlarını üretirler. Bu çıkışlar bir sonraki ara katman için girişler olacaktır. Bu girişler tekrar ağırlıklandırılarak bias değerleri ile toplanacak ve bu sinaptik cevaplar tekrar bir transfer fonksiyonundan geçirilerek bir sonraki katman için girişler veya ağıın çıkışı/ları haline gelecektir. Gizli katmanlar ve bu katmanlardaki işlemci eleman sayıları genellikle deneme yanılma yoluyla bulunur. Makul bir hata ile yüksek doğrulukta bir öğrenmenin sağlandığı durumlar denenerek ağ yapısı belirlenir. Çıkış katmanındaki işlemci eleman sayıları ise problemin türüne göre belirlenebilir [105, 106].

MLP ağlarda hatanın azaltılması amacıyla her bir işlemci eleman için ağırlık değerleri değiştirilerek ağın çıkışının arzu edilen değere yaklaşması sağlanır. Ağın çıkış hatasının azaltılması amacıyla güncellenen ağırlık değerleri bir önceki katmanda değiştirilerek çıkışa etkisi incelenir. Bu işlem ağın hatasının istenilen değere kadar azaltılması veya maksimum deneme sayısına ulaşılması durumlarında sonlandırılır. Ağırlık değerlerinin çıkış seviyesinden giriş katmanına doğru, veri akışının tersi yönünde güncellenmesi sebebiyle bu yönteme hatanın geriye yayılması (back propagation) algoritması denir [106].

4.4.1.1.2. Geri Beslemeli Yapay Sinir Ağları

Geri beslemeli (recurrent) yapay sinir ağları, en az bir işlemci elemanın çıkışının d gecikmesi ile kendi veya başka bir işlemci elemana giriş olarak uygulanması ile elde edilen ağ yapısıdır [107]. Geri beslemeli YSA yapıları doğrusal olmayan dinamik bir davranış sergiler. Her hangi bir andaki çıkış değeri, o andaki ve d süre önceki giriş değerlerine göre belirlenir. Önceki giriş verilerinin saklanması, ağın dahili bir hafıza oluşturmasını ve ağ girişlerine uygulanan veri dizilerinin bu hafıza ile yorumlanmasını sağlar. Dinamik yapılarından dolayı geri beslemeli yapay sinir ağları, tahmin uygulamalarında başarılı bir şekilde kullanılmaktadır [108]. Şekil 4.8’de ağın çıkışı girişlerine geri beslenmiş bir geri beslemeli ağ yapısı verilmiştir. Hopfield, SOM (Self Organization Map), Kohonen, ESN (Echo State Network), Elman ve Jordan ağları bu yapılara örnek verilebilir [92, 104, 107].



Şekil 4.8. Geri beslemeli YSA.

4.4.1.2. YSA'ların Öğrenme Yöntemlerine Göre Sınıflandırılması

Gözlem ve eğitimin doğal yapıda meydana getirdiği davranış değişikliği öğrenme olarak ifade edilir [107]. Bu tanımdan hareketle YSA'larda öğrenme, bir başka deyişle gözlem ve eğitim sonucu meydana gelen davranış değişiklikleri, ağırlıkların bir takım metod ve kurallar yardımıyla değiştirilmesi ile sağlanabilir. Öğrenme esnasında ağırlık, uygulanan veri setine uygun cevaplar üretmesi için ağırlıkların değiştirilmesi gerekir. Ağırlık bu davranış uyarlaması sırasında ağırlıklar genel olarak üç öğrenme metodu ile güncellenir. Danışmanlı, danışmansız ve takviyeli öğrenme metodları ve bunların uygulandığı değişik öğrenme kuralları mevcuttur.

4.4.1.2.1. Danışmanlı Öğrenme Metodu

Danışmanlı öğrenme metodunda, giriş verilerinin ve bu verilere karşılık gelen cevapların bulunduğu eğitici giriş-çıkış veri seti kullanılarak, ağırlık cevabı çıkış veri setiyle tanımlanan arzu edilen değerlere yaklaştırılarak ağırlıkların güncellenmesi sağlanır. Danışmanlı öğrenme metodu arzu edilen çıkışları üretmek amacıyla bir danışmana ihtiyaç duyar. Ağırlıkların güncellenmesi ve eğitimin tamamlanması için bir öğrenme kuralı seçilerek, ağırlık cevabı arzu edilen sonuçlara yaklaştırılır ve eğitim tamamlanır. Widrow&Hoff [87] ve Rumelhart&McClelland [109] tarafından geliştirilen delta kuralı veya geri yayılım algoritması danışmanlı öğrenmeye örnek olarak verilebilir.

4.4.1.2.2. Danışmansız Öğrenme Metodu

Danışmansız öğrenme metodunda, giriş verilerine karşılık gelen eğitici çıkış verileri ağırlık uygulanmaz. Ağırlık sadece giriş verileri uygulanır ve her bir giriş verisi için ağırlık kendi öğrenme kuralını geliştirerek sınıflandırma yapar. Ağırlıklar her bir sınıftan girişleri ayırtılabilecek şekilde hesaplanır ve öğrenme tamamlanır. Carpenter ve Grossberg tarafından geliştirilen Adaptif Rezonans Teorisi (ART) [110] ve Khonen tarafından geliştirilen SOM [92] yapılarında danışmansız öğrenme metodu kullanılmaktadır. Örüntü sınıflandırma, hedef tanıma ve sinyal işleme uygulamalarında kullanılırlar [111].

4.4.1.2.3. Takviyeli Öğrenme Metodu

Takviyeli öğrenme metodu danışmanlı öğrenme metoduna benzer bir yöntemdir. Fakat bu yöntemde öğrenme işleminde, arzu edilen çıkışların bilinmesine gerek yoktur. Bu yöntemde ağırlık girişler uygulanır ve bu verilerin ürettikleri cevabın giriş verisi ile ne

kadar uyumlu olduğu araştırılır. Çıkışlar ile girişler arasındaki uyumu incelemek amacıyla bir kriter kullanılır. Öğrenme gerçek zamanlıdır ve ağırlıklar deneme yanılma yoluyla bulunur. Optimizasyon problemlerini çözmek için Hinton ve Sejnowski [112] tarafından geliştirilmiş Boltzmann kuralı ve genetik algoritma takviyeli öğrenme metdolarına birer örnek olarak verilebilir.

4.4.2. YSA Öğrenme Algoritmaları

Yapay sinir ağlarının eğitiminde değişik ağ yapıları ile birlikte kullanılan çok sayıda öğrenme algoritması bulunmaktadır. MLP YSA'ların eğitiminde kullanılan öğrenme algoritmalarına örnek verilecek olursa; Bayes Düzleştirme (BR), Conjugate Gradyent with Powell/Beale (CGB), Fletcher-Powell Conjugate Gradyent (CGF), Polak-Ribiere Conjugate Gradyent (CGP), Esnek Geri Yayılım (RBP), Scaled Conjugate Gradyent (SCGB), Grup Momentumlu Geri Yayılım (BPM), Eşleştirmeli Gradyent Azaltma Geri Yayılım (EGBP), Delta Bar Delta (DBD), Genişletilmiş DBD (EDBD, Extendent DBD), Quasi Newton (QN) ve Levenberg & Marquardt (LM) algortimaları olarak sıralanabilir [113, 114].

MLP ağların eğitiminde çok sayıda öğrenme algoritması kullanılmasına rağmen bu yöntemlerden en çok kullanılanı LM geri yayılım algoritmasıdır [106, 115]. Bu tez çalışmasında en başarılı sonuçları üreten geri yayılım ve LM algoritmalarının kısa açıklamaları aşağıda verilmiştir.

4.4.2.1. Geri Yayılım (BP) Algoritması

Geri yayılım (BP, Back Propagation) algoritması MLP türü ağların eğitiminde en çok tercih edilen algoritmadır [106, 113, 115, 116]. Hatanın ağ boyuca, çıkış katmanından giriş katmanına doğru ağırlıkların değiştirilmesi ile azaltılması sebebiyle geri yayılım algoritması ismini almıştır.

BP algoritması i ve j katmanındaki nöronlar arasındaki ağırlık farkının k 'ıncı iterasyondaki değeri, $\Delta w_{ij}(k)$, Denklem 4.5'teki gibi hesaplanır.

$$\Delta w_{ij}(k) = \eta \delta_j x_i \quad (4.5)$$

Burada η öğrenme katsayısı, δ_j ise j nöronuna ait bir faktördür ve Denklem 4.6'daki gibi hesaplanır.

$$\delta_j = \frac{\partial f}{\partial net_j} (O_{dj}^k - O_j) \quad (4.6)$$

Burada O_{dj}^k , j işlemci elemanın arzu edilen çıkış değeridir. net_j Denklem 4.7'de verilmiştir.

$$net_j = \sum x_i w_{ij} \quad (4.7)$$

Arzu edilen çıkış değeri, O_{dj}^k , ancak ağın çıkış cevabı için tanımlanabilir. Gizli katmanlardaki işlemci elemanlar için herhangi bir arzu edilen çıkış değeri olmadığı için gizli katmanlardaki işlemci elemanların δ_j faktörü çıkış katmanından başlanarak giriş katmanına doğru, bütün işlemci elemanlar için Denklem 4.8'deki gibi güncellenir.

$$\delta_j = \frac{\partial f}{\partial net_j} \sum w_{qj} \delta_q \quad (4.8)$$

4.4.2.2. LM Algoritması

Levenberg & Marquardt (LM) öğrenme algoritması BP algoritmasının farklılaşmış bir türüdür. Bu öğrenme algoritması iyi bir yakınsama sağlar ve öğrenme süresi kısadır. LM algoritması Gauss&Newton (GNA) ve Gradient Descent algoritmalarının kısıtlamaları ortadan kaldırılarak, bu algoritmaların iyi özelliklerinin birleştirilmesi ile oluşturulmuştur. Bu algoritmalarından daha başarılı sonuçlar üretir, global minimum noktasının uzaklığı ve başlangıç değerlerine bağlı olarak GNA'dan daha uzun sürelerde sonuç üretebilir [113, 116].

LM algoritması X_k değerini her bir iterasyon için Denklem 4.9'daki gibi işletir. Denklemden p_k arama doğrusu, β ise arama doğrusu boyunca performansı azaltacak şekilde seçilen bir faktördür.

$$X_{k+1} = X_k + \beta \cdot p_k \quad (4.9)$$

Newton eşleştirmeli türev yöntemleri sonuca daha hızlı yakınsama sağlamasına rağmen, Hessian matrisinin hesabındaki zorluklar yüzünden tercih edilmezler. İleri beslemeli YSA'ların birçoğunda olduğu gibi, performans fonksiyonunun karelerin toplamı şeklinde yazılabildiği durumlarda Hessian matrisi Denklem 4.10'daki gibi tahmin edilir.

$$H = J^T J \quad (4.10)$$

Burada J Jakobian matrisidir ve standart geri yayılım algoritmasına göre hesaplandığı için Hessian matrisinden daha kolay hesaplanır. Hessian matrisinin türevinin hesaplanmasında Denklem 4.11'deki yaklaşım kullanılır.

$$g = J^T e \quad (4.11)$$

Burada e hata değeridir. Parametrelerin yakınsamasında Denklem 4.12'deki yaklaşım kullanılır.

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \quad (4.12)$$

Burada eğer μ değeri büyükse minimuma yaklaşma adımı küçültülür.

5. BÖLÜM

VERİ SETİNİN HAZIRLANMASI VE YSA MODELLERİNİN OLUŞTURULMASI

Genellikle YSA'nın (Yapay Sinir Ağı) boyutu büyüdükçe öğrenmesi daha iyi olur. Fakat aynı durumun test aşamasındaki performansı iyileştirdiği her zaman garanti edilemez. YSA'nın girişlerine uygulanan öznitelik seti ile ağın büyüklüğü arasında bir ilişki vardır. Büyük boyutlu bir ağ ve çok miktardaki giriş verisi tek başlarına yüksek performanslı bir YSA'yı garanti edemez. Bu bakımdan iyi hazırlanmış bir veri seti YSA eğitiminde oldukça önemlidir [117]. Fazla sayıdaki öznitelik, YSA girişine uygulanan veri sayısını arttırdığından işlemci elemanların ve ağın giriş sayılarının da artmasına sebep olur. İşlemci elemanların giriş sayısı arttıkça, kullandığı çarpıcıların da sayısı artmakta ve işlemci elemanların daha fazla kaynak kullanmasına sebep olmaktadır. İşlemci elemanların daha çok kaynak kullanması ve ağda kullanılan işlemci elemanların sayısının artması, ağın karmaşıklığının artmasına sebep olur. Bunun sonucu olarak da YSA'nın donanım gerçekleştirilmesi zorlaşmaktadır [33].

Bu tez çalışmasında 205, 208, 210 ve 215 numaralı MIT-BIH (MIT-Belt İsrail Hastanesi) kayıtlarından alınan ham veri dosyalarına TBA (Temel Bileşen Analizi) öznitelik çıkarım algoritması uygulanıp, veri setinin boyutu azaltılarak sınıflandırıcı olarak kullanılan MLP tipi YSA'nın girişlerine uygulanmıştır. Ham veri dosyalarında bulunan EKG kayıtlarındaki kalp atımlarına TBA yöntemi uygulandığında, her bir kalp atımı korelasyonu düşük veriler ile daha az sayıda öz nitelik ile temsil edilebilmektedir. Korelasyonu düşük olan veriler ile oluşturulmuş, düşük boyutlu yeni veri seti, ağın doğruluğunu çok fazla düşürmeden, YSA'nın boyutunu etkili bir şekilde azaltılmaktadır.

Üçüncü ve son adımda ise ağın N tipi verileri aşırı öğrenmesini ve diğer iki sınıftan kalp atımlarında (V ve F) hatalar yapmasını önlemek amacıyla, V tipi atımların yaklaşık 6 katı, F tipi atımların ise yaklaşık 12 katı olan 9.188 adet normal (N) atımının %80'i ikincil veri setinden rastgele çıkarılmıştır. Bu işlem veri setinde bir denge oluşturularak ağın N sınıfından atımları aşırı öğrenmeye gitmesini önlemek, dolayısıyla PVC doğruluğunu arttırmak amacıyla yapılmıştır. Tablo 5.1'de verildiği gibi nihai veri seti N ve V tipi atımları bir birlerine yakın örnek sayılarında ifade edecek şekilde düzenlenmiştir.

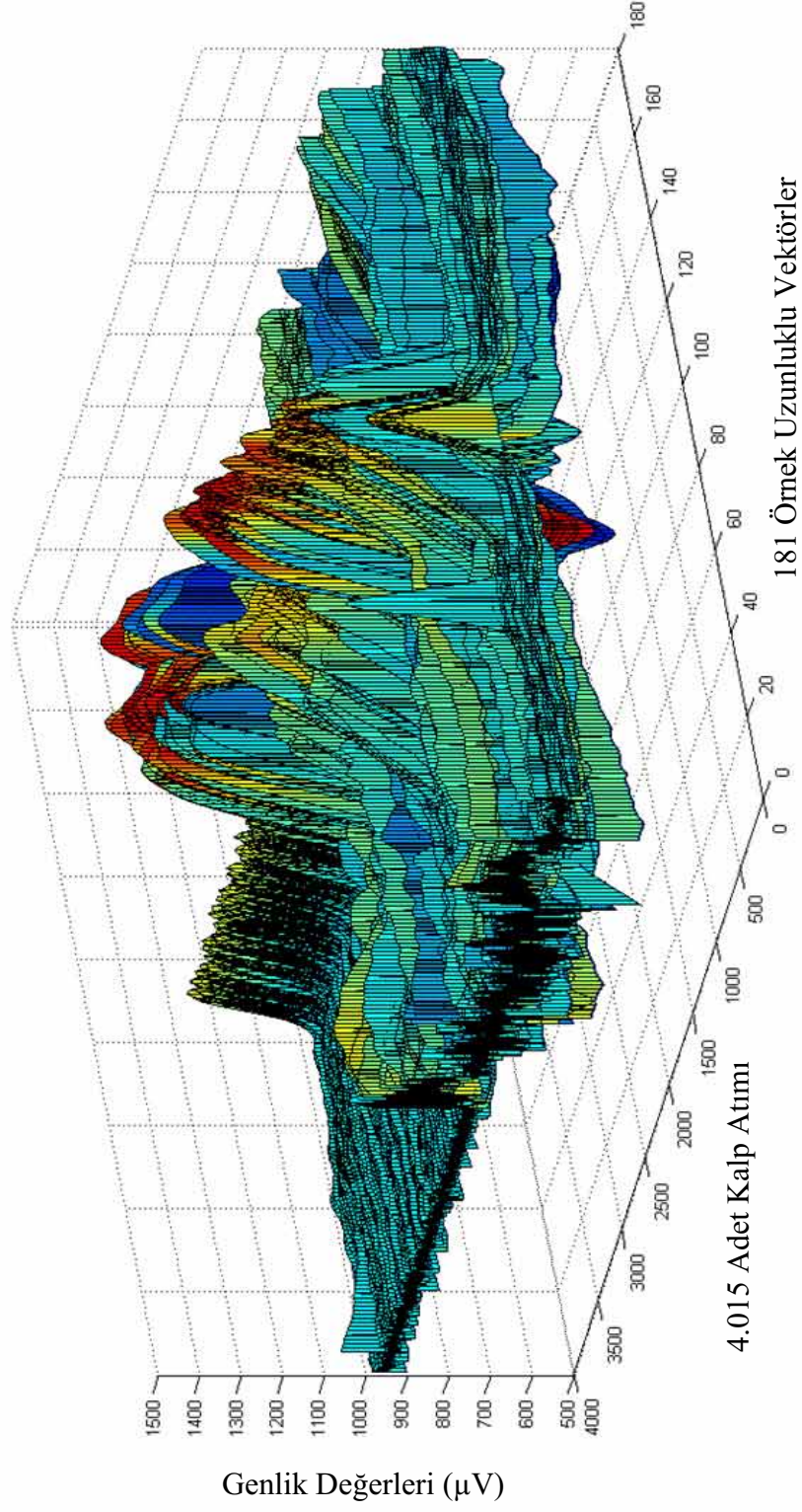
Tablo 5.1. Eğitim veri setlerinin oluşturulması.

	Aritmi Sınıfı	N	V	F	Toplam
1. Adım	Ham Veri Dosyaları*	9.221	1.477	756	11.454
2. Adım	İkincil Veri Seti	9.188	1.422	749	11.359
3. Adım	Nihai Veri Seti	1.844	1.422	749	4.015

Ham Veri Dosyaları 205, 208, 210 ve 215 kayıtların N, V ve F içeriklerinin tümüdür.*

Nihai veri setinin içeriği Şekil 5.7'de gösterilmiştir. Görüldüğü gibi nihai veri seti içerisindeki kalp atımları, genliği [500 ile 1.500] aralığında değişen bir nümerik sahaya sahiptir. Nihai veri setine normalizasyon işlemi uygulanarak, verinin değişkenliği [-2 ile 2] nümerik sahasına çekilmiştir. Bu sayede hem kalp atımları 0 çizgisi üzerine yerleştirilerek simetrik eksen kullanılmış hem de nümerik saha küçültülerek YSA eğitilirken kullanılan doğrusal olmayan aktivasyon fonksiyonunun satüre olduğu durumların sayısı azaltılmıştır.

Şekil 5.7'de 0-180 ekseni, 181 örnek vektörü uzunluğundaki kalp atımlarının her bir örneğini, 0 - 4.000 ekseni ise 4.015 adet kalp atımından oluşan, nihai veri setindeki her bir kalp atımını ifade etmektedir. 0 - 4.000 ekseninde ilk 749 veri F, 2.171 ile 750 arasındaki 1.422 veri V ve 2.172 ile 4.015 arasındaki 1.844 veri ise N sınıfından olayları temsil eder.



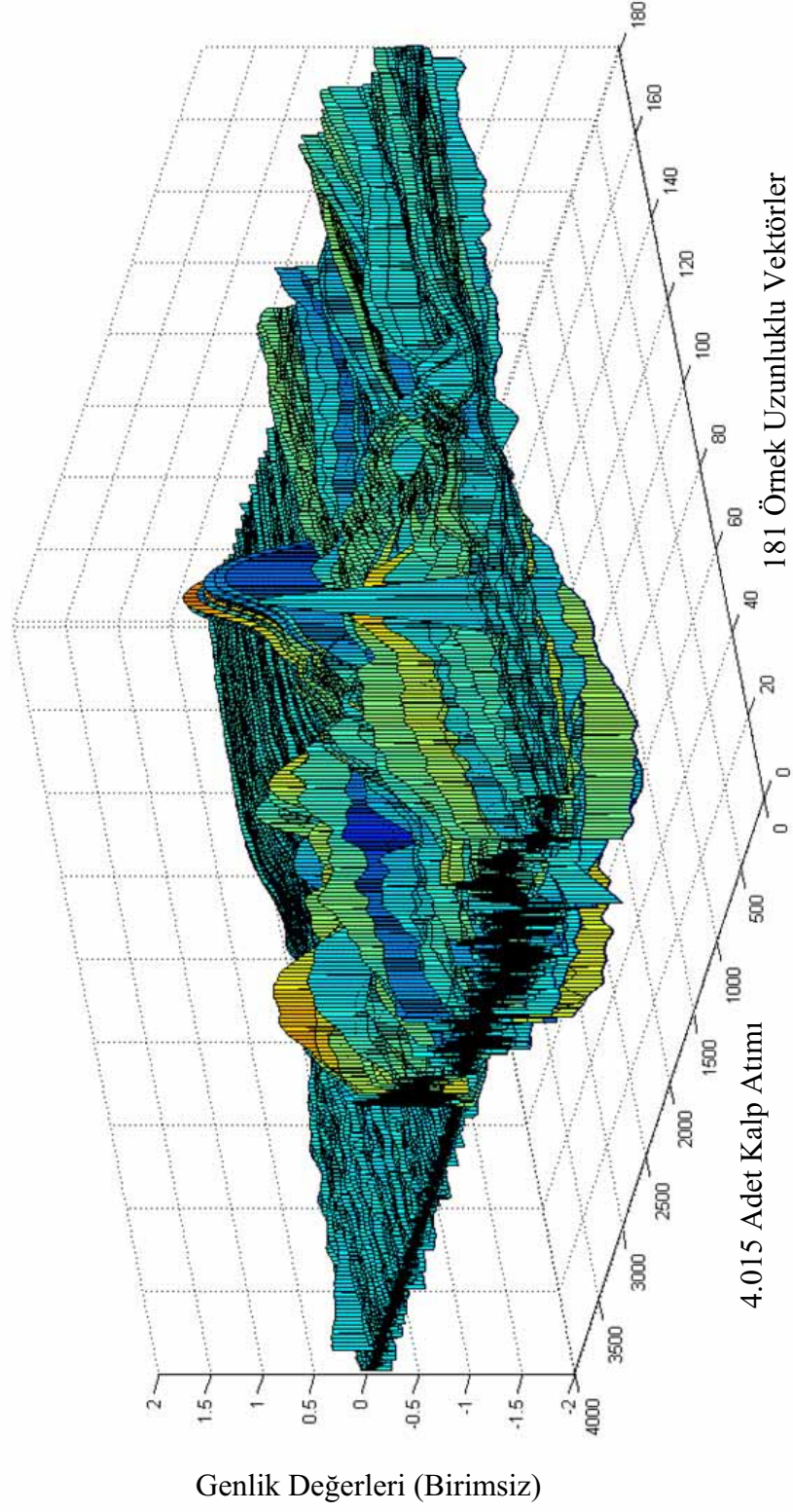
Şekil 5.7. Nihai veri seti.

Matlab içerisinde koşturulan normalizasyon algoritması aşağıdaki gibidir.

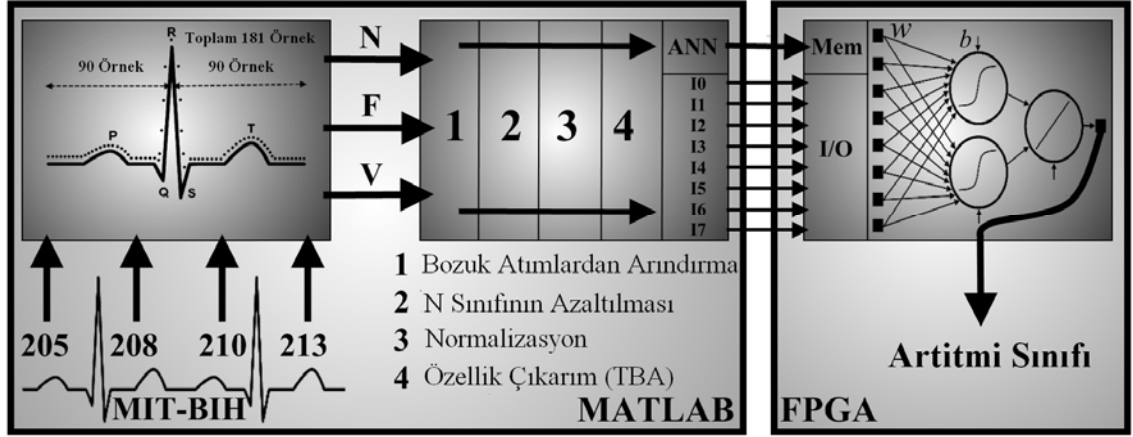
```
for
    i = 1:size(raw_data,2); % nihai veri setindeki kalp atım sayısı toplamı
    col_data = raw_data(:,i);
    mean_col = mean(col_data); % 181 örnek uzunluklu her bir atım için ortalama
    std_col = std(col_data); % her bir atımın standart sapması
    norm_data(:,i) = (col_data - mean_col)/(5*std_col); % normalize veri seti
end
```

Normalizasyon işlemi nümerik sayı aralığını [500 ile 1.500]'den [-2 ile 2] aralığına çekerek işareti sayısal olarak ifade etmekte kullanılacak olan ifadenin kısaltmasına yardımcı olacaktır. Nümerik aralığın daralması, nümerik tanımlama yönteminde kullanılacak bit sayısını azaltmakta ve çarpma işlemi sonucunda oluşacak taşmaları önlemektedir. Normalizasyon işlemi aynı zamanda ağın eğitim süresini ve hesaplama karmaşıklığını da azaltmaktadır. Normalizasyon işlemi sonunda elde edilen yeni veri setine, yani normalize veri setine TBA uygulanacaktır. Normalizasyon algoritması, terslendirilerek normalize veri setine tekrar uygulanırsa başlangıçtaki nümerik değerler tekrar elde edilebilir. Bu işlem sonunda elde edilen verilerin, her sınıftan atımları daha belirgin ve basit bir şekilde temsil ettiği görülmektedir. Normalize veri seti Şekil 5.8'de verilmiştir. Burada Şekil 5.7'de olduğu gibi 0-180 eksen, 181 örnek vektörü uzunluğundaki kalp atımlarının her bir örneğini, 0 - 4.000 eksen ise toplam 4.015 adet kalp atımından oluşan her bir kalp atımını ifade etmektedir. 0 - 4.000 ekseninde ilk 749 veri F, sonraki 1.422 veri V ve kalan 1.844 veri ise N tipi atımları ifade eder.

Artık veri setine TBA yöntemi uygulanacak durumdadır. Gelineen noktada normalize veri setine TBA yöntemi uygulanacak ve bu verilerden çıkarılan özniteliklerle YSA, Matlab ortamında eğitilecek ve elde edilen ağırlıkla YSA, FPGA yongası üzerinde donanımsal olarak gerçekleştirilecektir. Bu işlemlerin hangi geliştirme platformlarında ne sırayla gerçekleştirileceği Şekil 5.9'daki akış diyagramında verilmiştir.



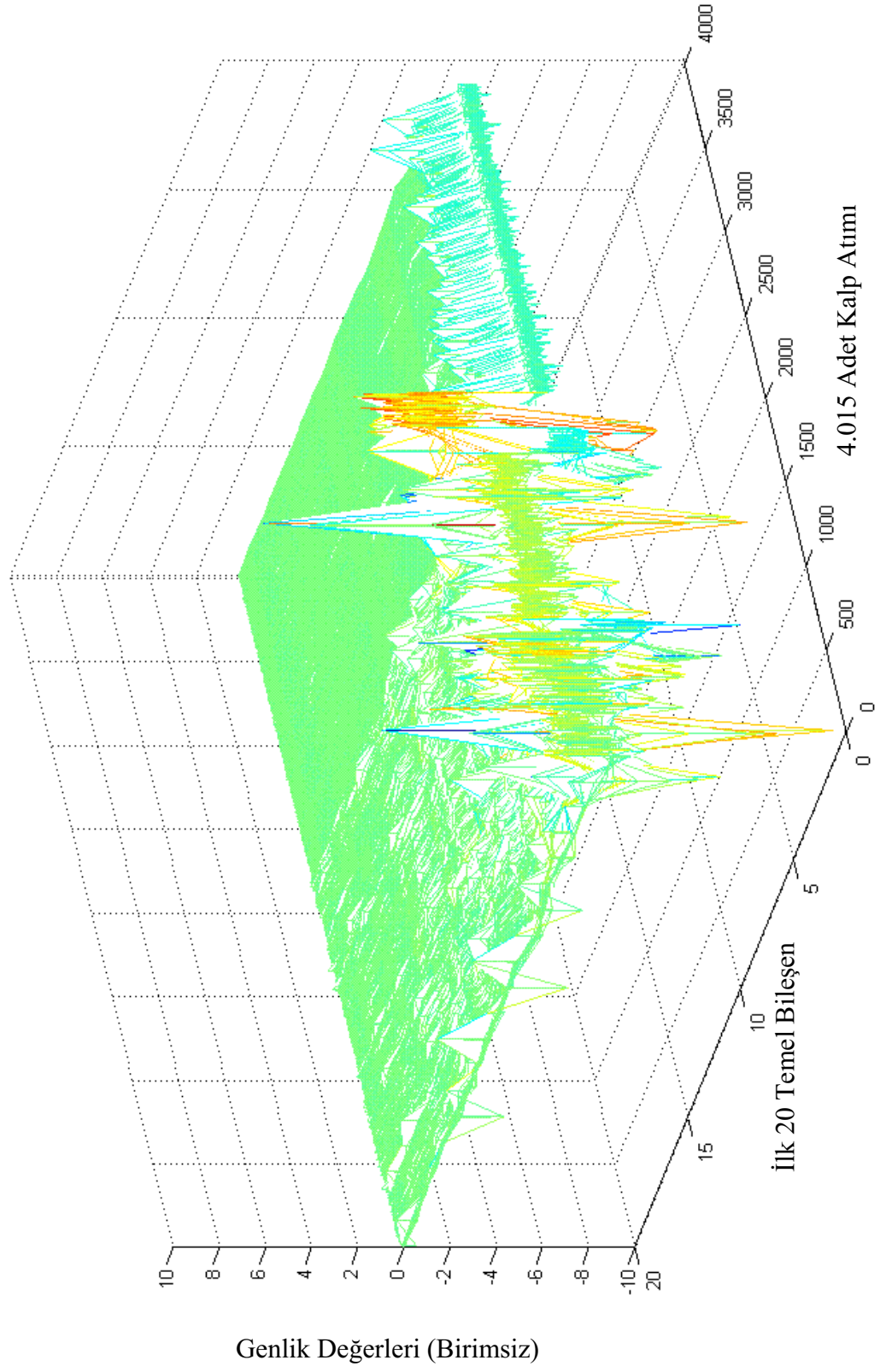
Şekil 5.8. Normalize veri seti.



Şekil 5.9. YSA tabanlı PVC sınıflandırıcının tasarım akış şeması.

5.2. Veri Setine TBA Uygulanması

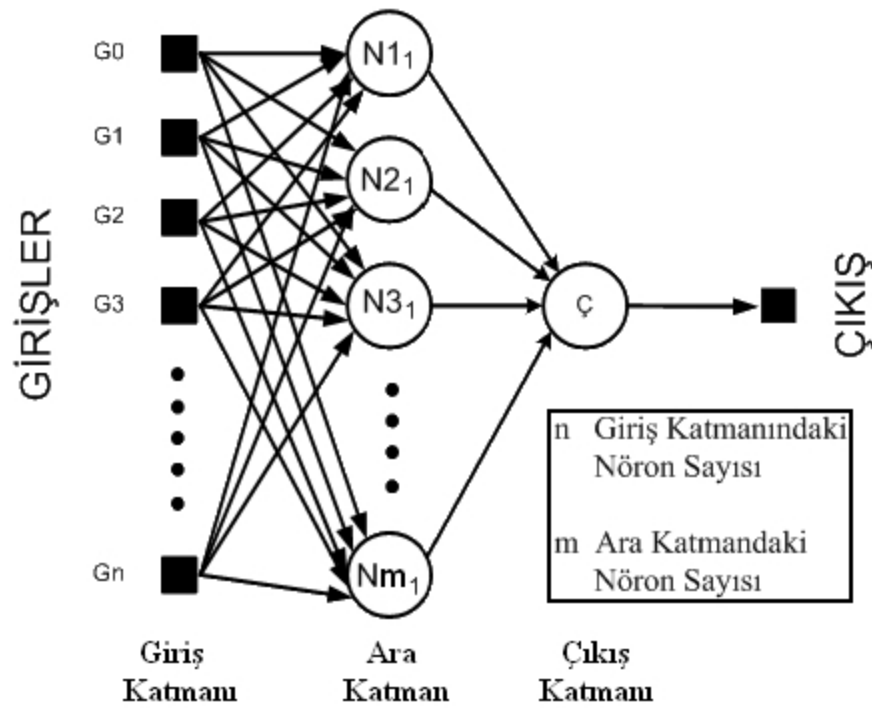
Normalize veri setine TBA yöntemi uygulanarak, 181 örnek vektörü uzunluğundaki her bir kalp atımını daha az sayıda veri ile tanımlayan temel bileşenler bulunur. Bu tez çalışmasında normalize veri setine TBA uygulanarak ilk yirmi temel bileşen bulunmuş ve değişik sayılarda (5, 6, ..., 20) temel bileşenin giriş olarak uygulandığı değişik boyutlardaki YSA'ların eğitiminde kullanılmıştır. Şekil 5.10'da verilen grafikte ilk yirmi temel bileşen için değişimin V ve F sınıfından veriler üzerinde yoğunlaştığı, N sınıfından veriler için temel bileşenlerdeki değişimin çok az olduğu görülmektedir. Bunun anlamı N sınıfından kalp atımlarını temsil etmek için gerekli olan temel bileşen sayısının, V ve F sınıfından kalp atımlarını ifade etmek için gereken temel bileşen sayısından daha az olduğudur. 181 örnek vektörü uzunluğundaki 4.015 kalp atımı, 726.715 (181×4.015) örnek uzunluğunda bir veriyi ifade eder. Normalize veri seti sekiz temel bileşen ile ifade edildiğinde, 4.015 kalp atımı için 32.120 (8×4.015) örnek uzunluğunda bir veri setine dönüşür. Böylece normalize veri seti $726.715 / 32.120 = 22,6$ kat azaltılarak, hesaplama yükü çok büyük ölçüde giderilmiştir. MIT-BIH veritabanı kayıtları 360 örnek vektörü uzunluğundadır. Bu verinin sekiz temel bileşen ile ifade edilmesi ile gerçekte 45 ($360/8$) kat veri indirgemesi sağlanmıştır. 45 kat büyüklüğünde bir verinin oluşturacağı hesaplama yükü düşünülecek olursa, ağır eğitim süresinde nasıl bir iyileştirme yapıldığı anlaşılacaktır. 4.015 atım için verilen eksende ham veri dosyaları için verilen sıra geçerlidir.



Şekil 5.10. İlk yirmi temel bileşen.

5.3. MLP Tipi Yapay Sinir Ağının Eğitilmesi

Bu çalışmada TBA analizi kullanılarak değişik örnek uzunluğundaki vektörlerden oluşan veri setleri üretilmiş ve bu veriler ile değişik giriş/ara katman işlemci elemanlarına sahip ağlar oluşturulmuştur. Oluşturulan bu ağlar değişik öğrenme algoritmaları ile eğitilmiştir. Şekil 5.11’de ağdaki değişkenlik, giriş işlemci eleman ve gizli katman işlemci eleman sayıları cinsinden verilmiştir. Maksimum giriş işlemci eleman sayısı $n = 20$, maksimum gizli katman işlemci eleman sayısı ise $m = 35$ ’tir.



Şekil 5.11. MLP YSA ağ değişkenliği.

Bir giriş katmanı, bir gizli katman ve bir çıkış katmanına sahip olan MLP mimarisi, giriş katmanına uygulanan öznitelik sayısı 2 ile 35 arasında ve gizli katmanda bulunan işlemci eleman sayıları ise 5 ile 20 arasında değiştirilerek elde edilen 544 değişik ağ modeli, farklı öğrenme algoritmaları kullanılarak eğitilmiştir.

Eğitimlerde kullanılan parametreler ve veri seti ortaktır. Eğitimlerde kullanılan veri seti bütün algoritmalar için nihai veri setidir. Eğitimlerde kullanılan hata algoritması karesel farkların ortalaması yöntemidir. Arzu edilen hata değeri e^{-6} ve maksimum iterasyon sayısı 1.500 olarak seçilmiştir. Hatanın ardı ardına 20 değer için doğrulanması istenilmiştir.

Bu tez çalışmasında giriş katmanında doğrusal olmayan sigmoid aktivasyon fonksiyonu bloğuna, çıkış katmanında ise doğrusal purelin aktivasyon fonksiyonu bloğuna sahip olan 544 adet değişik boyutlu ağ bir dizi öğrenme algoritması kullanılarak eğitilmiş ve en iyi performansın elde edildiği algoritma tespit edilmiştir. Bu ağların eğitimleri sırasında kullanılan öğrenme algoritmaları CGB (Conjugate Gradyent with Powell/Beale), CGF (Fletcher-Powell Conjugate Gradyent), CGP (Polak-Ribiere Conjugate Gradyent), SCG (Scaled Conjugate Gradyent), BPR (Resilient Back Propagation) ve LM (Levenberg Marquardt) algoritmalarıdır.

Matlab, ileri beslemeli ağlar için standart öğrenme algoritması olarak LM algoritmasını koşturur. Çünkü genellikle LM en hızlı öğrenme algoritmasıdır. Bununla birlikte bütün öğrenme algoritmalarında, ağın boyutu büyüdükçe artan ağırlık değerleri yüzünden eğitim süreleri uzamaktadır. Artan ağırlık değerlerinin güncellenmesi için daha fazla hafıza alanı ve daha uzun hesaplama zamanı gerekmektedir. Büyük boyutlu ağların eğitimi için CGP ve BPR öğrenme algoritmalarının kullanımı daha uygundur. Bunun sebebi bu algoritmaların hesaplamalar için ihtiyaç duydukları hafıza alanının nispeten küçük ve eğitim sürelerinin diğer algoritmalara nazaran daha kısa olmasıdır [114].

Geri yayılım (BP), YSA eğitimlerinde bazen özel olarak gradyent descent algoritmasını tarif etmek amacıyla kullanılır. Bununla birlikte çok katmanlı ağlar, bazen BP YSA olarak tanımlanabilir. Fakat gradyent ve Jakobian matrislerinin hesaplamasında kullanılan BP yöntemi, değişik birçok ağın eğitiminde de kullanılabilir [114]. Bu tez çalışmasında BP terimi, hatanın geriye yayılımı ile yapılan hesaplamayı ifade eder ve bu çalışmada kullanılan bütün öğrenme algoritmaları BP tabanlıdır. Bu yüzden geri yayılım terimi yerine, her bir öğrenme algoritması için kendi özel ismi kullanılmıştır.

Tablo 5.2’de CGB, Tablo 5.3’te CGF, Tablo 5.4’te CGP, Tablo 5.5’te SCG, Tablo 5.6’da BPR ve Tablo 5.7’de LM algoritmaları ile eğitilmiş 544 adet farklı ağ yapısı test hataları ile birlikte verilmiştir. 544 farklı YSA konfigürasyonunun test hatalarının ortalamaları CGB için %3,54, CGF için %3,09, CGP için %3,73, SCG için %3,83, BPR için %2,53 ve LM için %2.17 değerindedir. Sonuçta LM öğrenme algoritması ile daha yüksek sınıflandırma başarımları elde edildiği gözlemlenmiştir [118].

Tablo 5.2. CGB algoritması ile eğitilmiş 544 adet değişik yapıdaki ağların test hataları.

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

TB* YSA girişine uygulanan öznitelik sayılarının her biri için hata ortalaması,

NS* Gizli katmandaki nöron sayılarının her biri için hata ortalaması,

%3,54 değeri 544 farklı boyutlu ağların ürettiği hataların ortalamasıdır.

Tablo 5.3. CGF algoritması ile eğitilmiş 544 adet değişik yapıdaki ağların test hataları.

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

TB* YSA girişine uygulanan öznitelik sayılarının her biri için hata ortalaması,

NS* Gizli katmandaki nöron sayılarının her biri için hata ortalaması,

%3,09 değeri 544 farklı boyutlu ağların ürettiği hataların ortalamasıdır.

Tablo 5.4. CGP algoritması ile eğitilmiş 544 adet değişik yapıdaki ağların test hataları.

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

TB* YSA girişine uygulanan öznitelik sayılarının her biri için hata ortalaması,

NS* Gizli katmandaki nöron sayılarının her biri için hata ortalaması,

%3,73 değeri 544 farklı boyutlu ağların ürettiği hataların ortalamasıdır.

Tablo 5.5. SCG algoritması ile eğitilmiş 544 adet değişik yapıdaki ağların test hataları.

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

TB* YSA girişine uygulanan öznitelik sayılarının her biri için hata ortalaması,

NS* Gizli katmandaki nöron sayılarının her biri için hata ortalaması,

%3,83 değeri 544 farklı boyutlu ağların ürettiği hataların ortalamasıdır.

Tablo 5.6. BPR algoritması ile eğitilmiş 544 adet değişik yapıdaki ağların test hataları.

Hata		Temel Bileşen Sayısı																	NS*
		5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
Gizli Katmandaki Nöron Sayısı	2	3,34	3,01	3,16	3,09	3,29	2,57	2,54	2,49	1,89	2,59	2,19	2,29	2,72	2,42	2,14	1,99	2,61	
	3	3,71	3,01	3,31	2,54	2,49	3,11	2,07	2,14	2,09	2,44	2,39	2,19	2,27	3,09	2,09	1,92	2,56	
	4	3,26	2,99	2,54	3,46	3,11	2,32	2,64	2,82	2,02	2,57	2,32	2,69	2,04	2,34	3,01	2,32	2,65	
	5	3,31	3,84	3,11	2,29	2,32	2,37	2,44	2,67	2,02	2,34	2,42	2,24	2,24	2,22	1,92	2,22	2,50	
	6	2,86	2,72	3,14	2,79	2,27	2,72	2,77	2,49	2,64	2,17	1,79	2,37	1,82	2,14	1,92	2,39	2,44	
	7	3,44	3,24	2,89	3,19	2,59	2,14	2,37	2,17	2,34	2,86	2,09	2,19	2,19	2,32	2,24	1,82	2,51	
	8	2,94	3,81	2,69	2,39	2,22	2,94	2,37	2,69	2,69	2,24	2,22	2,39	2,09	1,92	2,62	2,24	2,53	
	9	2,96	3,86	2,91	2,37	2,29	2,39	2,44	2,49	2,32	2,32	2,27	2,12	2,54	2,27	2,19	2,22	2,50	
	10	3,56	3,06	2,59	3,44	2,59	2,12	2,59	2,57	2,64	2,54	1,99	1,97	2,27	1,94	1,99	1,99	2,49	
	11	2,82	3,09	2,59	2,52	2,14	3,19	3,19	2,02	2,94	2,54	2,37	1,97	2,12	2,04	2,27	1,97	2,49	
	12	3,61	3,84	3,69	2,39	2,49	2,47	2,44	2,22	2,42	2,02	2,22	2,12	2,77	1,99	2,07	2,02	2,55	
	13	4,29	3,59	3,06	3,06	1,92	2,64	2,59	2,04	2,37	2,29	1,94	2,14	2,12	2,12	2,44	1,99	2,54	
	14	3,26	3,74	3,14	2,27	2,64	2,29	2,57	2,09	2,07	2,12	2,39	2,02	2,57	2,27	2,02	2,04	2,47	
	15	2,77	2,94	3,36	3,34	2,32	2,37	2,34	2,69	2,49	2,47	2,22	1,94	2,22	1,99	2,02	2,44	2,49	
	16	2,77	3,19	2,64	2,39	2,37	2,27	2,79	2,22	2,67	2,02	1,82	2,12	2,02	2,14	2,14	2,04	2,35	
	17	3,49	2,99	2,84	2,72	2,32	2,27	2,44	2,64	2,29	2,37	2,34	2,24	1,97	2,44	2,52	2,12	2,50	
	18	3,51	2,94	9,17	2,57	2,42	2,24	2,52	2,72	2,59	1,84	2,89	1,97	2,09	1,84	2,17	2,29	2,86	
	19	3,56	3,79	2,52	3,49	2,22	2,91	2,49	2,57	2,37	2,09	1,94	2,19	2,14	2,12	1,92	1,97	2,52	
	20	5,06	3,69	2,77	2,37	3,66	2,34	2,07	2,74	2,39	2,14	2,29	2,74	2,14	2,17	2,09	2,12	2,67	
	21	3,06	3,34	2,74	2,49	2,42	2,17	2,27	2,59	2,79	2,22	1,84	1,77	2,17	2,17	2,19	1,89	2,38	
	22	4,29	3,24	2,64	2,49	2,52	2,32	3,24	2,54	2,59	2,42	2,19	2,29	2,19	1,97	2,17	2,07	2,57	
	23	3,46	3,54	2,69	2,24	2,74	2,34	2,42	2,34	2,04	2,29	2,39	2,17	1,87	1,97	2,09	2,09	2,42	
	24	3,36	3,34	2,89	2,32	2,24	2,84	2,44	2,44	2,69	2,62	2,04	2,39	2,19	2,12	2,24	2,07	2,51	
	25	4,63	3,54	3,31	2,44	2,69	2,54	2,57	2,42	2,39	2,32	2,86	2,91	2,12	2,27	2,44	1,77	2,70	
	26	2,67	3,14	3,69	2,64	2,24	2,37	2,34	2,27	2,32	2,12	2,69	2,14	2,09	1,99	1,89	2,09	2,42	
	27	3,84	3,09	2,67	3,61	2,72	2,94	2,19	2,09	2,12	1,97	2,34	2,12	2,02	2,02	2,29	2,24	2,52	
	28	2,59	3,61	3,24	3,06	2,24	3,89	2,29	2,37	2,64	2,19	2,24	2,04	1,97	2,27	2,37	2,12	2,57	
	29	2,82	3,66	2,94	2,47	2,52	2,57	2,94	2,64	2,59	1,94	2,04	3,11	2,44	2,19	1,99	2,02	2,56	
	30	3,74	3,19	2,64	2,47	2,29	2,69	2,49	2,59	1,84	2,02	2,14	2,67	2,52	2,44	1,97	2,14	2,49	
	31	3,11	2,99	3,31	2,82	2,67	2,62	2,59	2,22	2,07	2,19	2,24	2,04	2,04	1,62	2,07	2,67	2,45	
	32	3,24	2,77	2,42	2,27	2,47	2,47	3,24	2,32	2,34	3,09	2,07	2,07	2,14	2,29	3,04	2,39	2,54	
	33	3,06	2,99	2,96	3,16	2,34	2,49	2,77	2,42	2,47	2,42	2,24	2,17	2,44	2,02	2,14	2,12	2,51	
	34	3,19	3,21	2,67	2,99	2,32	2,49	2,37	2,89	2,07	2,22	1,84	1,97	1,89	2,34	2,17	2,54	2,45	
	35	2,84	2,96	3,11	2,64	3,66	2,12	2,44	2,91	2,04	1,92	2,19	2,64	2,39	2,34	2,34	2,52	2,57	
	TB*	3,37	3,29	3,12	2,73	2,52	2,54	2,54	2,46	2,36	2,29	2,22	2,25	2,20	2,17	2,21	2,14	2,53	

TB* YSA girişine uygulanan öznitelik sayılarının her biri için hata ortalaması,

NS* Gizli katmandaki nöron sayılarının her biri için hata ortalaması,

%2,53 değeri 544 farklı boyutlu ağların ürettiği hataların ortalamasıdır.

Tablo 5.7. LM algoritması ile eğitilmiş 544 adet değişik yapıdaki ağların test hataları.

Hata		Temel Bileşen Sayısı																NS*
		5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Gizli Katmandaki Nöron Sayısı	2	4,01	2,79	2,84	2,34	2,12	2,37	1,99	2,02	2,04	1,72	1,57	2,04	2,02	2,07	1,77	1,77	2,22
	3	3,09	3,04	3,71	2,47	2,29	1,99	1,99	2,34	1,99	2,02	1,69	2,02	1,84	1,69	1,87	1,67	2,23
	4	2,96	2,94	2,64	2,54	2,34	1,82	2,44	2,86	1,77	1,72	2,47	1,79	1,57	2,02	2,12	1,74	2,23
	5	3,09	2,69	2,67	2,91	2,14	2,12	2,24	1,87	2,02	1,79	1,87	1,64	1,99	2,04	1,92	1,79	2,17
	6	2,62	2,77	2,64	2,39	2,49	2,37	2,17	2,12	2,24	1,92	1,99	1,69	1,47	1,67	2,04	1,67	2,14
	7	3,04	2,64	2,29	2,27	2,09	2,17	2,24	1,79	1,97	2,49	1,97	1,87	1,87	1,87	2,22	1,77	2,16
	8	2,79	2,54	2,44	2,39	2,07	1,82	2,02	2,24	1,59	1,77	1,74	1,84	2,14	2,07	2,39	1,92	2,11
	9	2,91	3,64	2,52	3,04	2,44	2,47	1,82	2,07	1,89	1,87	2,14	1,97	2,14	1,97	1,77	1,59	2,27
	10	2,72	2,57	2,54	2,22	2,47	2,47	2,04	2,19	1,69	1,74	1,84	1,99	2,12	1,94	1,74	1,72	2,13
	11	3,29	2,59	2,79	2,12	2,09	1,92	2,07	1,89	2,17	1,74	1,92	2,09	1,84	1,72	1,87	1,74	2,12
	12	2,91	2,64	2,27	2,24	2,86	2,14	2,14	1,44	2,12	1,89	1,64	2,04	2,24	1,84	1,74	1,92	2,13
	13	3,26	2,72	2,74	2,09	2,19	1,94	2,17	2,04	1,99	1,74	1,84	1,67	2,09	1,92	2,07	1,72	2,14
	14	2,59	2,74	2,59	2,52	2,42	2,32	1,99	2,27	1,84	1,89	1,97	1,84	1,44	2,42	1,99	1,97	2,18
	15	2,86	2,54	2,49	2,37	2,07	1,92	1,94	2,22	1,77	1,82	1,64	2,07	2,34	1,97	1,94	1,64	2,10
	16	2,84	2,77	2,74	2,17	2,29	2,04	1,89	2,34	2,52	2,32	1,69	1,89	2,34	2,19	2,12	1,84	2,25
	17	2,32	3,16	2,96	2,91	1,92	2,27	2,12	1,82	1,87	1,67	2,02	1,77	1,92	1,77	1,72	1,54	2,11
	18	2,77	2,99	2,64	3,04	2,32	2,32	1,97	2,27	1,87	2,02	2,04	1,74	1,54	1,89	1,89	1,87	2,20
	19	5,38	2,69	3,19	2,86	2,14	2,44	2,09	2,14	2,74	1,69	2,27	1,52	2,24	1,84	2,12	2,14	2,47
	20	2,67	2,69	2,69	2,02	2,29	2,32	2,29	1,87	2,19	1,87	2,04	2,02	1,89	2,17	2,07	2,04	2,20
	21	3,31	2,86	2,59	2,24	2,02	1,82	1,87	1,89	1,97	1,67	1,49	1,92	1,77	1,77	2,32	1,77	2,08
	22	2,74	2,54	2,34	2,86	2,12	2,22	2,22	1,77	2,17	1,97	1,72	1,79	1,77	1,92	2,29	2,19	2,16
	23	2,49	2,44	2,74	2,54	2,32	2,42	2,02	2,27	2,44	2,04	2,02	1,59	1,79	1,84	2,04	1,67	2,17
	24	2,86	2,44	2,27	2,52	2,77	1,84	2,09	2,17	2,04	1,64	1,94	1,59	2,02	1,87	1,92	1,44	2,09
	25	2,77	2,86	2,59	2,47	2,14	1,72	2,04	1,99	2,27	2,04	2,44	1,82	1,77	1,92	2,04	1,92	2,18
	26	3,24	2,54	2,42	2,47	1,94	2,02	2,19	1,99	2,22	1,94	1,97	2,44	1,89	1,94	1,64	2,04	2,18
	27	3,29	2,79	2,42	2,34	2,19	1,87	2,02	2,17	1,89	1,79	1,97	1,72	1,79	1,72	1,99	1,74	2,10
	28	2,74	2,79	2,34	2,02	1,89	2,02	1,82	1,74	2,04	1,89	1,84	1,99	2,07	1,82	1,92	1,99	2,06
	29	2,77	2,44	2,34	2,09	2,09	2,39	1,97	2,19	1,94	2,09	1,97	1,92	1,74	1,82	2,04	1,94	2,11
	30	2,64	2,94	3,61	2,19	1,84	2,49	2,07	1,89	2,07	2,54	2,12	1,89	1,72	2,04	2,07	2,09	2,26
	31	3,06	2,42	2,52	2,54	2,09	2,07	2,34	1,72	1,72	2,07	1,84	2,24	2,02	2,02	1,79	1,82	2,14
	32	2,94	3,01	2,49	2,12	2,14	1,97	2,29	1,97	1,79	1,62	1,94	2,24	1,74	1,64	1,79	1,62	2,08
	33	3,54	2,72	2,44	2,39	2,09	2,22	2,57	2,22	2,57	1,99	2,14	1,59	2,02	1,84	2,22	2,12	2,29
	34	2,52	2,57	2,52	2,54	2,34	1,77	2,09	1,67	2,02	2,19	2,86	1,89	2,02	1,84	1,62	1,97	2,15
	35	3,19	2,84	2,62	2,42	2,14	1,99	2,12	2,64	1,92	1,82	1,67	1,74	1,92	1,64	1,82	1,82	2,14
	TB*	3,01	2,75	2,64	2,43	2,21	2,12	2,10	2,06	2,04	1,91	1,95	1,88	1,91	1,90	1,97	1,83	2,17

TB* YSA girişine uygulanan öznitelik sayılarının her biri için hata ortalaması,

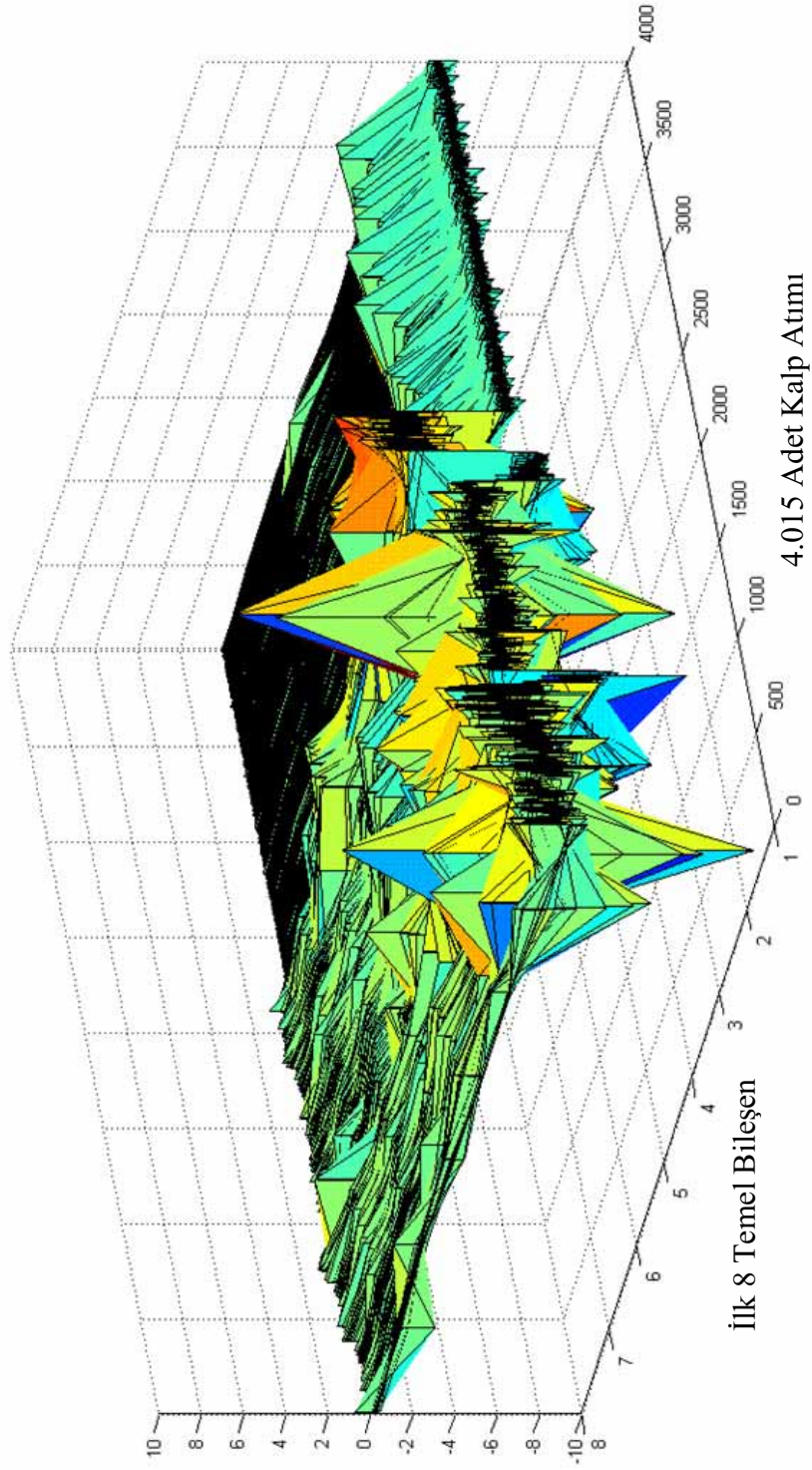
NS* Gizli katmandaki nöron sayılarının her biri için hata ortalaması,

%2,17 değeri 544 farklı boyutlu ağların ürettiği hataların ortalamasıdır.

LM algoritması ile eğitilmiş Tablo 5.7'deki değişik boyutlu YSA'larda ortalama hata değeri %2,17 iken, giriş katmanında 20, gizli katmanda 35 ve çıkış katmanında 1 işlemci eleman bulunan 20x35x1 boyutlu ağı hata %1,82 ve 5x2x1 boyutlu ağı hata %4,01 değerindedir.

Tablo 5.7 dikkatle incelendiğinde, YSA girişlerine uygulanan özniteliklerin yani TBA ile üretilen temel bileşen (TB) sayısının, YSA hatası üzerine etkisi açık bir şekilde görülmektedir. Bu değişim TB satırında verilmiştir. TB satırı YSA giriş sayısı sabit iken gizli katmandaki işlemci eleman sayılarının değişiminin YSA'nın hatası üzerine etkisini inceler. TB satırında her bir hücre, 34 gizli işlemci elemanının kombinasyonunun oluşturduğu hataların ortalamasıdır. TB satırı incelenecek olursa, YSA girişine uygulanan öznitelik sayısı 5 öznitelikten 20 özniteliğe çıkarılırken, YSA'nın hatası sırası ile %3,01 değerinden %1,83 değerine doğru azalmaktadır. Gizli katmandaki nöron sayısının YSA'nın hatası üzerine etkisinin incelenmesi için tabloda NS sütunu verilmiştir. Bu sütunun her bir satırı sabit sayıdaki gizli katman nöronu için YSA'nın hatasını vermektedir. Gizli katmandaki işlemci eleman sayısının, ağı performansına TB kadar büyük bir etkiye sahip olmadığı gözlemlenmiştir. Ara katmandaki işlemci eleman sayısı 2 adetten 35 adete doğru artarken, hatanın ağ ortalaması olan %2,17 civarında sabit kaldığı gözlenmektedir. Fakat gerek TB gerekse NS sayıları tek başlarına çok anlamlı değildir. Bu iki parametrenin birlikte yorumlanması gerekmektedir.

TB sütunu boyunca hata ortalamasının, beş temel bileşenden dokuz temel bileşene kadar artması esnasında hatanın önemli ölçüde azaldığı görülmektedir. Dokuz temel bileşenden yirmi temel bileşene doğru artan öznitelik sayıları, hatayı daha düşük adımlarla azaltmaktadır. Bu etki Şekil 5.10'da verilen yirmi temel bileşenin değişiminden de görülebilir. Öncül temel bileşenlerde, veriyi temsil eden değişim yüksekken, ikinci temel bileşenlerden itibaren verideki değişkenliğin azaldığı görülmektedir. Bu yüzden hem ağların ortalama hatasına yakın hem de temel bileşenlerin değişiminin nispeten yüksek olduğu, %2,34 hataya sahip küçük boyutlu 8x2x1 YSA modeli bu tez çalışması için referans model olarak seçilmiş ve 8x2x1 boyutlu YSA hem kayan noktalı hem de sabit noktalı nümerik tanımlamalar ile FPGA üzerinde gerçekleştirilmiştir. İlk sekiz temel bileşenin değişkenliği Şekil 5.12'de verilmiştir.



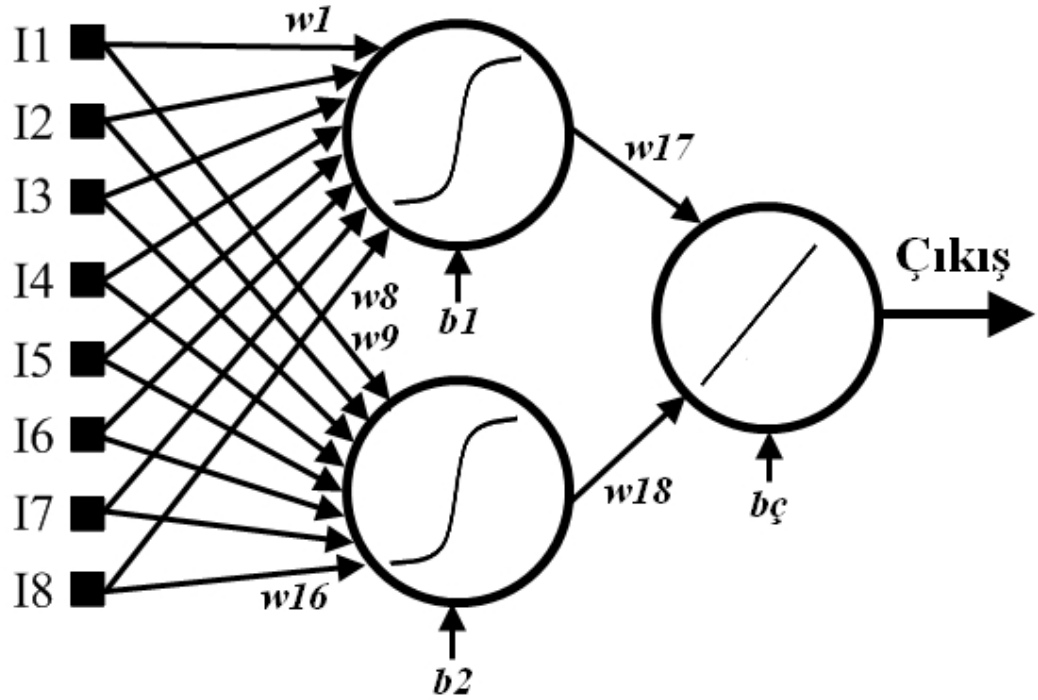
Genlik Değerleri (Birimsiz)

Şekil 5.12. İlk sekiz temel bileşen.

Aktivasyon fonksiyonun ağın doğruluğuna olan etkisini göstermek için Tablo 5.7’de en iyi sonuçları ürettiği tespit edilen LM öğrenme algoritması ile gizli katmanda doğrusal purelin aktivasyon fonksiyonu kullanılarak oluşturulmuş YSA modelleri eğitilmiş ve 8x2x1 boyutundaki ağ modelinin hatasının %22,3 olduğu tespit edilmiştir. Doğrusal purelin aktivasyon fonksiyonunun gizli katmanda kullanıldığı YSA yapıları, doğrusal olmayan sigmoid (logsig) fonksiyonunun kullanıldığı YSA modelleri kadar yüksek doğruluklarla eğitilememiştir. Doğrusal olmayan sigmoid aktivasyon fonksiyonunun gizli katmanda kullanıldığı YSA yapılarının ortalama hatası %2,17 iken, gizli katmanda doğrusal purelin aktivasyon fonksiyonunun bulunduğu YSA yapılarının ortalama hatası ise %20,1’dir. Aradaki bu fark aktivasyon fonksiyonlarının öğrenmeye olan etkilerini çok iyi bir şekilde göstermektedir.

5.4. 8x2x1 Boyutlu YSA’nın Ağırlıklarının Elde Edilmesi

8x2x1 boyutlu YSA, giriş katmanına uygulanan sekiz öznitelik yani ilk sekiz temel bileşen ile eğitilmiştir. 8x2x1 boyutlu YSA modeli sekiz giriş işlemci elemana sahip bir giriş katmanı, doğrusal olmayan logaritmik sigmoid aktivasyon fonksiyonuna sahip iki işlemci elemanı bulunan bir gizli katman ve doğrusal purelin aktivasyon fonksiyona sahip bir çıkış katmanından oluşmaktadır. Bu ağın giriş, çıkış ve gizli katmanındaki işlemci elemanların transfer fonksiyonu tipleri Şekil 5.13’te verilmiştir. Her bir giriş için ağırlık ($w1, w2, w3...w18$) ve bias ($b1, b2$ ve $bç$) değerleri de gösterilmiştir.



Şekil 5.13. 8x2x1 YSA modeli.

N, V ve F sınıfından veriler aşağıdaki kurala göre sınıflandırılırlar. Çıkış işlemci elemanında aktivasyon fonksiyonu olarak doğrusal purelin fonksiyonunun seçilmesinin sebebi de Denklem 5.1’de verilen bu çıkış kuralıdır. Denklemden görüldüğü gibi bu üç sınıftan veri doğrusal bir fonksiyon ile tanımlanabilir.

$$\text{Sınıf Kararı} = \begin{cases} F, & CIKIS \leq 1.5 \\ V, & 1.5 < CIKIS \leq 2.5 \\ N, & 2.5 < CIKIS \end{cases} \quad (5.1)$$

Ağın eğitimi tamamlandıktan sonra ağırlık ve bias değerleri, Matlab aktif work dizini içerisindeki net dosyasında saklanır. Matlab komut kod satırına net.b(1) yazılıp çalıştırıldığında, birinci gizli katmandaki iki işlemci elemana ait bias değerleri ($b1, b2$), net.b(2) çalıştırıldığında ise çıkış işlemci elemanının bias değeri ($bç$) elde edilir. Ağırlıkların elde edilmesi için net.IW{1}(1,:) komutu ile gizli katmandaki birinci nöronun ağırlık değerlerini ($w1, w2, w3, w4, w5, w6, w7$ ve $w8$), net.IW{1}(2,:) komutu ile gizli katmandaki ikinci işlemci elemanın ağırlık değerleri ($w9, w10, w11, w12, w13, w14, w15$ ve $w16$) elde edilir. Çıkış işlemci elemanının ağırlık değerleri ($w17, w18$) ise net.IW{2} komutu ile elde edilir.

Şekil 5.13'te blok diyagramı verilen, gizli katmanında doğrusal olmayan logsig aktivasyon fonksiyonu, çıkış katmanında ise doğrusal purelin aktivasyon fonksiyonu kullanılarak LM öğrenme algoritması ile eğitilmiş, 8x2x1 boyutlu YSA'nın %2,34 hata ile sonuç üreten bu modeline ait ağırlık ve biaslardan oluşan ağ parametreleri Tablo 5.8'de verilmiştir. Bu YSA modelinin hatası Tablo 5.7'de verilmişti.

Tablo 5.8. 8x2x1 boyutlu YSA için ağ parametreleri FPANN.

$w1$ = -0,6258	$w9$ = 147,1781	$w17$ = -0,9535
$w2$ = 23,4966	$w10$ = 84,5194	$w18$ = -1,0071
$w3$ = 43,9367	$w11$ = 11,9251	$b1$ = -47,8726
$w4$ = -6,6568	$w12$ = 55,6788	$b2$ = 36,3476
$w5$ = -56,1103	$w13$ = -54,9280	$bç$ = 0,9869
$w6$ = 10,6938	$w14$ = -54,0371	
$w7$ = -24,5478	$w15$ = 70,5659	
$w8$ = -53,1894	$w16$ = 117,5449	

6. BÖLÜM

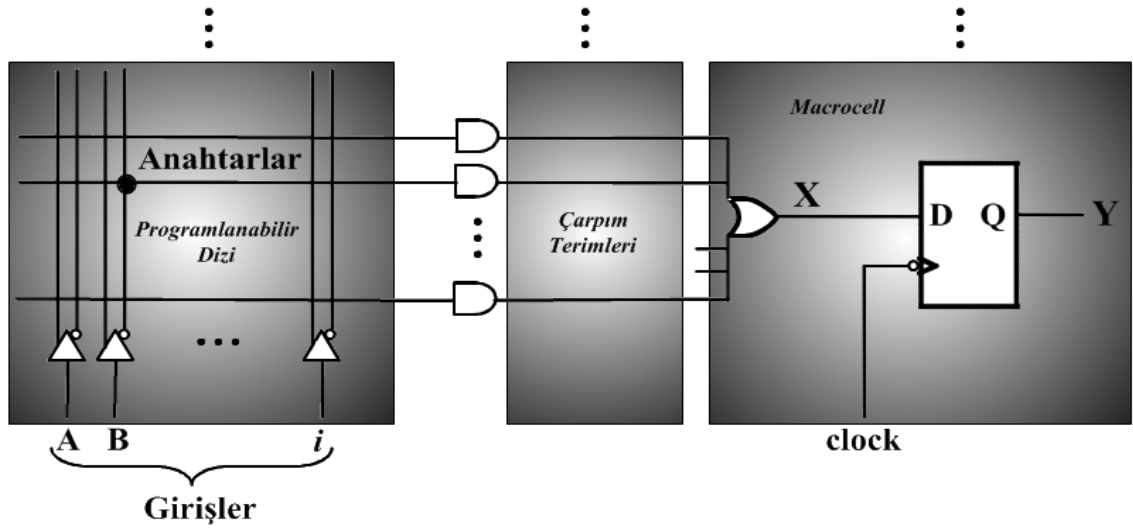
PVC TİPİ ARİTMİ SINIFLANDIRICININ FPGA DONANIMININ OLUŞTURULMASI

Programlanabilir Lojik (Programmable Logic), temel lojik fonksiyonları kullanarak kompleks donanım modelleri ortaya koymaya yarayan bir donanım geliştirme platformudur. Evdeki yüksek çözünürlüklü televizyondan, yakınlardaki GSM vericisine, bankadaki ATM makinesinden, araçtaki navigasyon cihazına kadar çok çeşitli alanlarda programlanabilir lojik kullanılmaktadır. Modern Kompleks Programlanabilir Lojik Devreler (CPLD, Complex Programmable Logic Devices) ve Alan Programlanabilir Kapı Dizileri'ne (FPGA, Field Programmable Gate Arrays) kadar gelişen programlanabilir lojik teknolojisi, Texas Instruments'ın 7400 serisi entegre devre elemanlarını bir Baskılı Devre Kartı (PCB, Printed Circuit Board) üzerinde birbirlerine bağlamakla başlar. 1960 ve 1970'li yıllarda birçok tasarım, içerisinde temel lojik kapılar, flip floplar ve bazı özel lojik yapıların bulunduğu 7400 serisi TTL (Transistör Transistör Lojik) entegre devre elemanları ile yapılmıştır [119].

TTL ile tasarımda amaç genellikle, mümkün olduğunca az sayıda entegre devre kullanarak maliyeti düşürmek ve devre kartının alanını küçültmektir. Devre kartı yapılmadan önce, tasarımcının devreyi kafasında planlamış olması gerekmektedir. Örneğin bir tasarımda ihtiyaç duyulan bir OR kapısı için fazladan entegre devre kullanmak yerine, var olan NAND kapılarını kullanarak daha az sayıda devre elemanı kullanılmış olacaktır. Bu ve benzeri optimizasyonların, lojik fonksiyonun cevabını değiştirmedikten emin olmak için devre gerçeklemesine geçilmeden önce tasarımın doğrulanması gerekmektedir. Doğruluk tablosu ve Karnough haritası çıkarılarak fonksiyon cevabı üretilmiş olan ifade, gerektiğinde Boolean denklemlerine başvurularak en az sayıda devre elemanı ile veya tasarımcının elinde bulunan devre elemanlarına göre son halini alır.

Birçok fonksiyon çarpımların toplamı şeklinde ifade edilebilir veya sadeleştirilebilir. Bu tür fonksiyonlar iki lojik seviyede, AND kapıları ile çarpımlar ve OR kapıları ile toplamalar olmak üzere, çarpımların toplamı fonksiyonu şeklinde gerçekleştirilebilir. TTL lojik devreler ile yapılan tasarımlarda, bu fonksiyonları icra eden entegre devrelerin devre kartı üzerinde birbirlerine bağlanması gerekmekteydi. Programlanabilir Lojik Devreler (PLD, Programmable Logic Devices) ile bu fonksiyonlar genelleştirilerek tek bir entegre devre şeklinde üretilmiştir. Programlanabilir Lojik Diziler (PLA, Programmable Logic Arrays) ilk PLD entegre devreleridir. PLA entegre devrelerinde hafıza elemanı bulunmamaktadır. Çıkışların birbirleri ile senkronizasyonu veya çıkış değerlerinin saklanması gerekmiyorsa, hafıza elemanının kullanılmasına gerek yoktur. Ancak çıkışın diğer çıkışlar ile senkronize olması veya saklanması isteniyorsa, bu durumda çıkışlar için birer flip flop yani hafıza elemanı gerekir. Şekil 6.1’de verilen şematik diyagramda, X çıkış değerine kadar olan kısım, iki lojik seviyeden oluşan PLA yapısını göstermektedir [120].

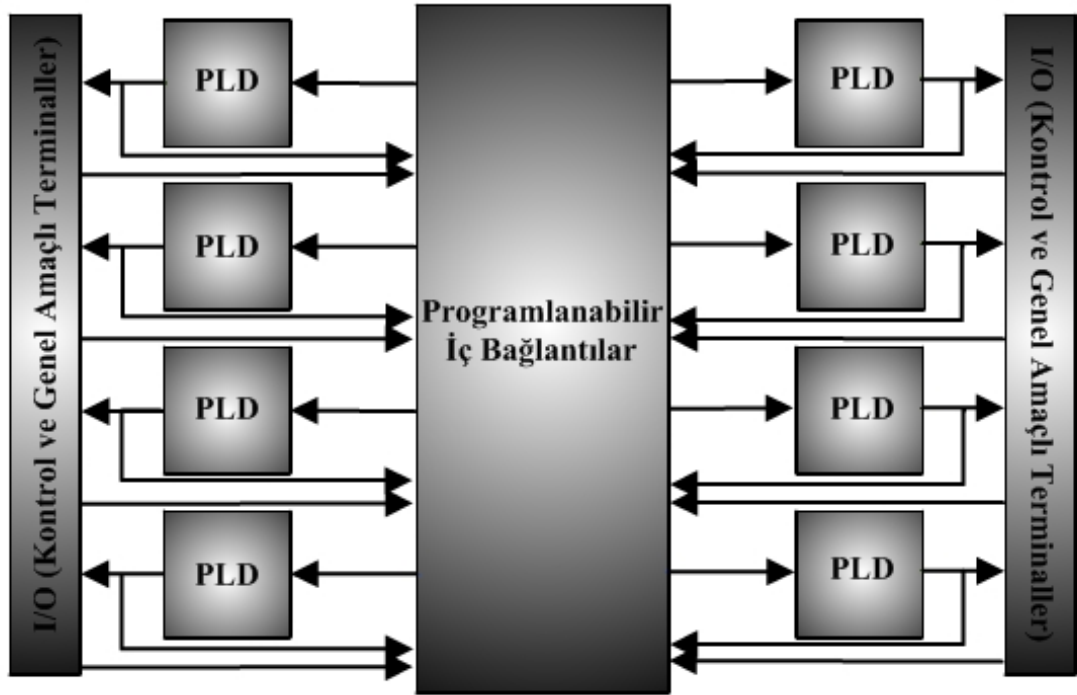
Advanced Micro Devices firması tarafından üretilen Programlanabilir Dizi Lojik (PAL, Programmable Array Logic) entegre devrelerinde, çarpımların toplamı fonksiyonu çıkışı (X), bir hafıza elemanına giriş olarak uygulanmıştır. Bu şekilde PLA mimarisi geliştirilmiş ve PLD entegre devrelerine senkronizasyon ve veri saklama özellikleri kazandırılmıştır. Çarpımların toplamı çıkışı ile hafıza elemanının birleştirildiği bu yapıya Macrocell denir. PLA’nın hafıza elemanı eklenerek geliştirilmesi sonucu ortaya konulan temel PAL mimarisi Şekil 6.1’de verilmiştir. Burada X kombinasyonel (combinational), Y ise sıralı (sequential) PAL çıkışları olarak ifade edilebilir. Şekil 6.1’de i adet giriş için bir çıkış satırı gösterilmiştir. Örneğin ilk nesil sıralı PAL entegre devrelerinden PAL16R8 entegresinde 8 giriş, 8 çıkış, bir clock ve bir OE (Output Enable, çıkışı aktif eden kontrol girişi) terminali vardır. Ayrıca Lattice Semiconductor firması tarafından üretilmiş, Türetilmiş Dizi Lojik (GAL, Generic Array Logic) isminde PLD’ler de bulunmaktadır. PAL ve GAL marka isimleridir ve üreticisinin PLD ürün ailesini temsil etmektedir. PAL ve GAL ürün ailelerinde, birbirlerine muadil çok sayıda PLD entegre devresi bulunmaktadır [119, 120].



Şekil 6.1. Temel PAL mimarisi.

PAL'lerin kullanımı ile lojik devre tasarımlarında pek çok avantaj sağlanmıştır. Bu entegre devrelerin içerisinde daha fazla sayıda lojik eleman bulunması sebebi ile tasarımlar daha az sayıda entegre devre ile gerçekleştirilebilmektedir. Daha az sayıda entegre devre demek, daha küçük bir devre kartı, daha düşük bir maliyet ve daha az enerji tüketimi anlamlarına gelir. Ayrıca az sayıda entegre devre, daha az bağlantı anlamına gelir. Bu ise arıza arama ve test işlemlerini kolaylaştırır. PAL ile sağlanan en önemli avantajlardan biri de tasarımın güvenliğidir. Geleneksel 7400 serisi entegre devreler ile yapılan bir tasarım, bir mühendis tarafından kolayca analiz edilip kopyalanabilir. Tasarımın bir PAL yongası içerisinde oluşturulması ile kopyalanması önlenmektedir [119, 121, 122].

PAL aynı zamanda çok büyük bir tasarım esnekliği de sunmaktadır. Tasarımcı istediği sayısal tasarımı, devre elemanı bulup bulamayacağını düşünmeden, tek tip entegre devreler ile gerçekleştirebilmektedir. Bu tasarım esnekliği tasarım yapmayı daha karmaşık bir hale soksa da, üreticilerin sunmuş olduğu geliştirme yazılımları, tasarım sürecini daha da basitleştirmiş ve kısaltmıştır. Bu entegre devrelerin çok önemli diğer bir avantajı ise programlanabilir olmalarıdır. Bu sayede hataların giderilmesi veya tasarımın geliştirilmesi için devre kartının ve üzerindeki entegrelerin değiştirilmesine gerek kalmamaktadır [103, 119].



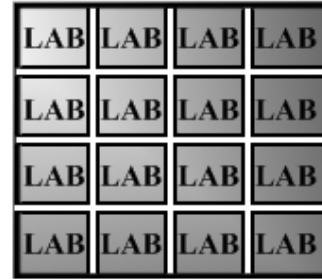
Şekil 6.3. Temel CPLD mimarisi.

CPLD'ler ile geleneksel PLD teknolojisi daha ileri bir noktaya taşınmıştır. Daha büyük ölçekli tasarımlar için daha yoğun CPLD yongaları üretilmiştir. Ancak bir tasarım ne kadar karmaşıklaştı ve iç bağlantıları ne kadar arttıysa, CPLD içerisindeki LAB'ları iyi organize edilmiş bir şekilde birleştirmek de o denli zorlaştı. Bu ise Merkezi Programlanabilir İç Bağlantılar bloğu üzerinden birbirleri ile bağlanabilecek maksimum LAB sayısında sınırlamalar meydana getirdi. Bu sınırlama, entegre devre üreticilerini merkezi bağlantı bloğu yerine satır ve sütunlar boyunca LAB'ların birbirlerine bağlanmasını mümkün kılacak yeni bir bağlantı tekniği oluşturmaya yöneltti. Şekil 6.4'te yüksek yoğunluklu bir CPLD içerisindeki LAB bloklarının, daha küçük bir alan içerisinde nasıl etkin bir şekilde bağlanabileceği gösterilmiştir. Şekilden de görüldüğü gibi LAB blokları tıpkı bir şehir planında olduğu gibi, her birini paralel ve dik olarak geçen bağlantı yolları arasına yerleştirilmişlerdir. Bu yerleşim tekniği sayesinde hücrelerin birbirleri ile bağlantıları daha etkili bir hale getirilmiştir. CPLD'lerde artan LAB sayısı bağlantı yollarının sayısını üstel bir şekilde arttırırken, yeni bağlantı tekniği ile bağlantı yolları daha etkin kullanılmış ve bağlantıların artışı doğrusal bir düzeye çekilmiştir. Bu yeni bağlantı tekniği FPGA yongaları içerisinde etkin bir şekilde kullanılmaktadır [119, 120, 122].



LAB sayısı arttıkça iç bağlantıların sayısı üstel olarak artar.

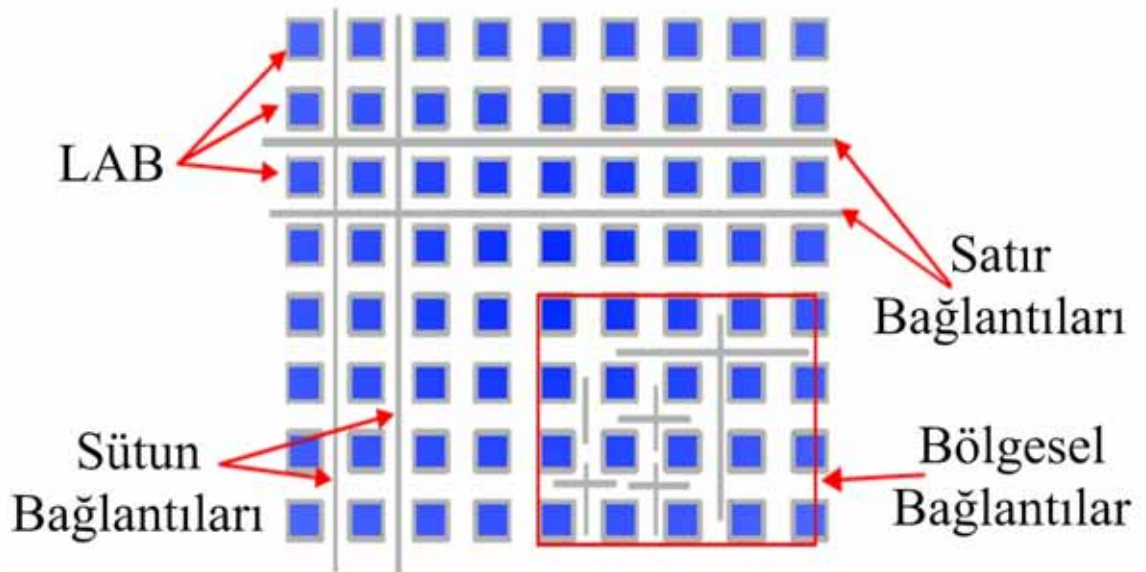
FPGA



LAB sayısı arttıkça iç bağlantıların sayısı doğrusal (linear) olarak artar.

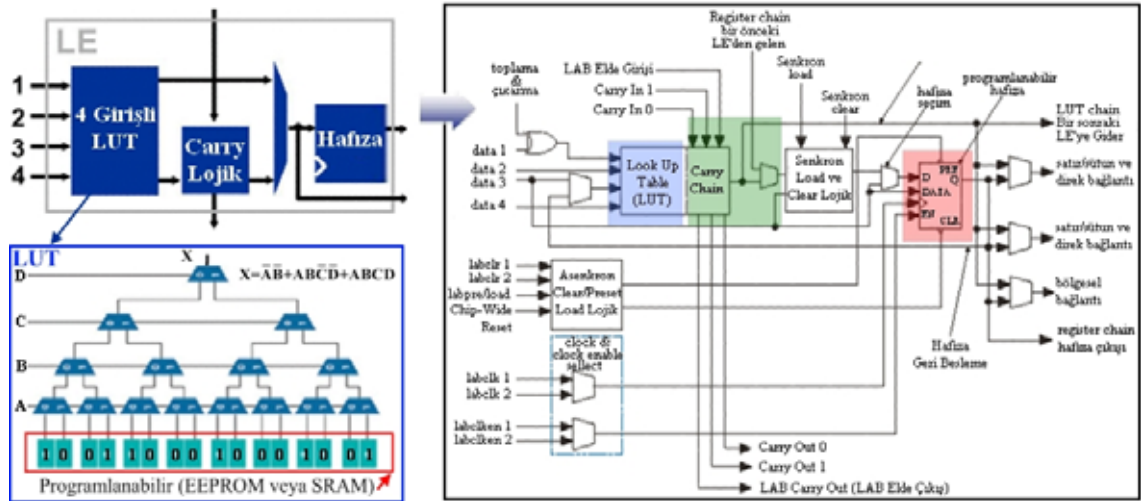
Şekil 6.4. İç bağlantılar.

FPGA yongasının içinde yer alan LAB'lar Lojik Eleman'lardan (LE, Logic Element) oluşur ve tıpkı CPLD'de olduğu gibi bütün hücreler yeniden programlanabilir özelliktedir. CPLD ile FPGA arasındaki farklılıklardan en göze çarpanları, merkezi bağlantı yolları yerine satır-sütunlardan oluşan ve istenilen herhangi bir bölgede bağlantıyı mümkün kılan yollar ve farklılaşmış LAB bloklarıdır. FPGA yongası içerisindeki bağlantı tekniği Şekil 6.5'te verilmiştir [119, 120, 123, 124].



Şekil 6.5. FPGA iç bağlantıları.

CPLD'ler içerisindeki LAB'lar programlanabilir diziler ve Macrocell'lerden oluşurken, FPGA içerisindeki LAB'lar bir dizi yeni özellik ile daha iyi konfigüre edilebilen ve kullanılmayan lojik kaynak sayısını azaltan, programlanabilir LE'lerden oluşur. Şekil 6.6'da bir LE, blok diyagram ve şematik gösterim olarak verilmiştir. LE'lerin bu avantajları FPGA yongalarını CPLD yongalarından daha kullanışlı hale getirmiştir. Her bir LE hücresi dört girişli bir giriş hafıza (LUT, Look Up Table), bir Elde Lojik (Carry Logic) ve bir çıkış hafıza bloğundan oluşur. LUT yöntemi ile fonksiyonlar kolayca üretilebilir ve saklanabilir bir hale gelmiştir. Ayrıca günümüz FPGA'ları gömülü çarpıcı, Faz Kilitli Döngü (PLL, Phase Locked Loop), gömülü hafıza, yüksek hızlı alıcı-verici blokları gibi gelişmiş birçok ilave donanım ile birlikte üretilmektedir [122, 123].



Şekil 6.6. LE içyapısı.

Üzerinde her türlü sayısal tasarımının oluşturulabileceği FPGA donanım geliştirme platformu, büyük ölçekli tasarımları hızlı ve ucuz bir şekilde ortaya koyabilme avantajı sağlar. Çünkü FPGA teknolojisi sayesinde her bir tasarım veya revizyon için yonganın tekrar üretilmesi zorunlu olmadığı gibi güçlü geliştirme yazılımları ile çok karmaşık tasarımlar dahi kısa sürelerde oluşturulabilmektedir. Her ne kadar Uygulamaya Özel Entegre Devrelerin (ASIC, Application Specific Integrated Circuit) yonga boyutu ve güç tüketimi FPGA'lardan çok daha küçük olsa da üretim süreçleri uzun ve maliyetlidir. FPGA yongalarında üretim süreç ve maliyeti, bilgisayar ortamında tasarlanan mimarinin sentezlenerek yonga üzerine yüklenmesi kadar kısa ve ucuzdur. Günümüzde içerisinde milyonlarca kapı elemanı olan FPGA yongaları üretilmektedir. Sonuç olarak FPGA yongaları, tasarım aşamasında veya az sayıda üretilecek olan ürünler için pratik bir uygulama geliştirme ortamı olmaktadır [125].

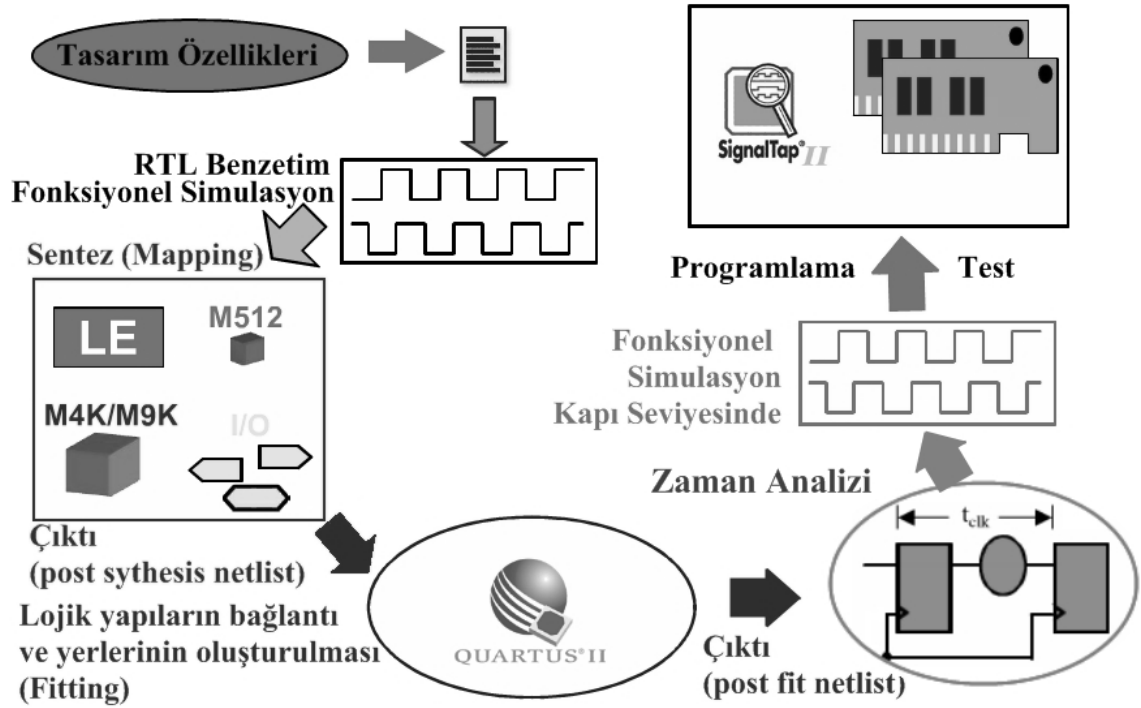
İşaret işleme uygulamalarında algoritmalar genellikle gelişmiş mikroişlemciler üzerinde gerçekleşir. Yüksek seviyeli özel amaçlı algoritmalar ise ya bir dizi Sayısal İşaret İşlemci (DSP, Digital Signal Processor) ya da ASIC yongalar üzerinde gerçekleşmektedir. FPGA teknolojisindeki gelişmeler sinyal işleme alanında çalışan araştırmacılara yeni yaklaşımlar sunmaktadır. FPGA yongasının uygulamalardaki temel avantajı, tıpkı bir ASIC gibi özel bir amaç için tasarlanabilir olmasının yanında, ASIC'ten farklı olarak düşük geliştirme ve üretim maliyetlerinin yanında daha sonra yapılabilecek değişiklik ve eklentilere de imkân sağlamasıdır. Ayrıca FPGA optimal devre gereksinimlerini gerçekleştirme hususunda, tasarım esnekliğinin yanı sıra, gerek PCB alanı gerekse güç tüketimi bakımından geleneksel DSP yongalardan daha iyidir. ASIC'e ihtiyaç duyulan uygulamalarda eğer pazara çıkma süresi veya tasarım uyumluluğu önemli birer parametre ise FPGA burada iyi bir çözüm olabilmektedir. FPGA kapı dizilerinin esnekliği ve performansı hızlı tasarım süreçlerini mümkün kılmaktadır. Tasarım yenilenmek veya değiştirilmek istenildiğinde, tasarımın gerçekleştirildiği yonga veya PCB değiştirilmek yerine yeniden programlanır [123].

FPGA'lar içerisinde konfigürasyon bilgisi saklamadıkları için, ilk enerjilendiklerinde harici bir EEPROM, CPLD veya işlemci gibi bir dış birim üzerinden konfigürasyon bilgilerinin okunarak programlanmaları gerekmektedir. SRAM türü çok hızlı hafızalar ile programlama süresi minimuma indirilerek bu dezavantaj ortadan kaldırılmıştır. Bir FPGA aktif ve pasif olmak üzere iki şekilde programlanabilir. Aktif programlamada FPGA enerjilendiği zaman, bir dış hafıza içerisindeki programlama dosyası FPGA tarafından okunarak programlama işlemi otomatik olarak tamamlanır. Pasif programlamada ise programlama süreci, işlemci gibi zeki bir çevresel donanım tarafından yapılır. İşlemci EEPROM'da veya başka bir konfigürasyon biriminde saklanan programın FPGA'ya nasıl ve ne zaman yükleneceğini kontrol eder. Ayrıca tasarım esnasında bilgisayarda oluşturulan tasarımların FPGA'ya yüklenmesi için JTAG arayüzü üzerinden bir kablo ile de programlanması mümkündür. Doğruluğu onaylanmış bir tasarım ASIC'e dönüştürülebilir, üretici firmalar adetli üretimler için konfigürasyonu sabitlenmiş FPGA'lar da üretmektedirler. Bu şekilde ASIC haline getirilmiş son ürünlerde hem birim adet başına maliyet düşer, hem de harici bir dış hafıza ve programlama donanımlarına ihtiyaç kalmaz. Bu da programlama için ihtiyaç duyulan süre sıkıntısını da ortadan kaldırır [119, 120, 123, 124].

6.1. FPGA ve Donanım Geliştirme Yazılımları

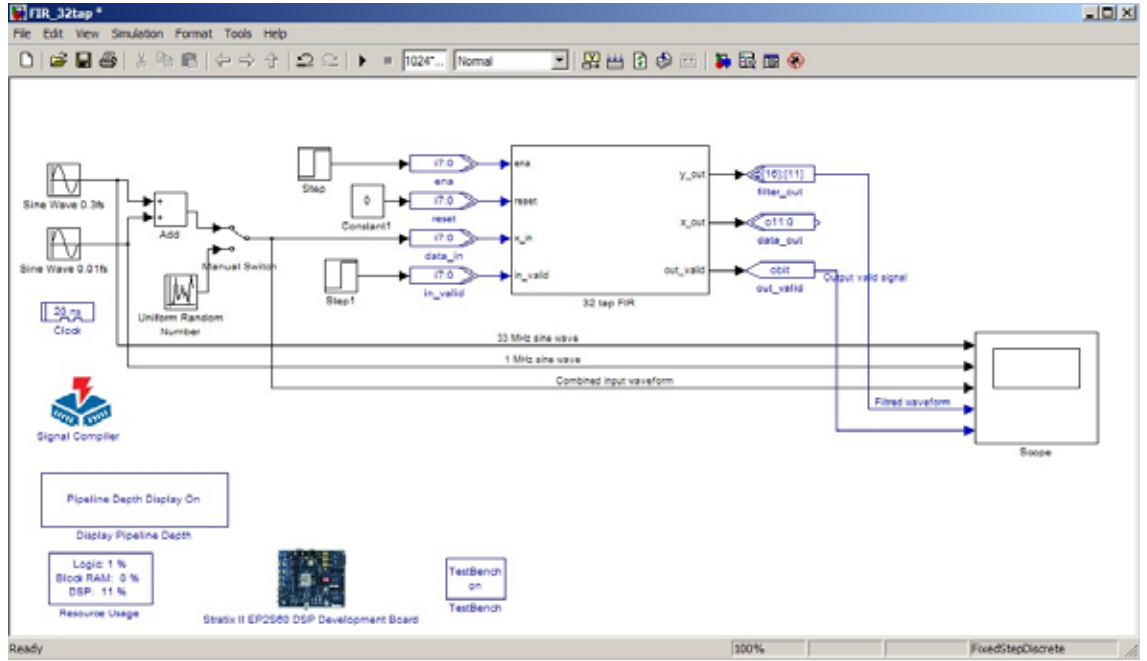
Bu tez çalışmasında Altera firmasının Cyclone III FPGA yongası kullanmıştır. Altera firmasının kendi ürettiği FPGA'ları için tasarımdan, programlama ve teste kadar bütün süreçlerinde kullanılabilen Quartus II isminde bir yazılımı bulunmaktadır. FPGA ile tasarım süreci, tasarım özelliklerini belirlemekle başlar. Tasarımın özelliklerine göre, davranışsal veya lojik yapılar, gerek bazı grafik ara yüzleri gerekse Donanım Tanımlama Dili (HDL, Hardware Description Language) kullanılarak Register Transfer Seviyesinde (RTL, Register Transfer Level) oluşturulur. Genellikle bu işlemin ardından Mentor Graphics veya ModelSim gibi yazılımlar kullanılarak oluşturulan yapının RTL fonksiyonel benzetimi yapılır. Fonksiyonel benzetim sadece lojik yapının doğruluğunu test etmektedir, FPGA içerisindeki zaman gecikmeleri bu benzetimde dikkate alınmaz. Bir sonraki adımda tasarım seçilen FPGA yongasının sahip olduğu donanım kaynak ve özelliklerine göre sentezlenir. Tasarım, sentez işlemi süresince arzu edilen alan, zaman ve performans kriterlerini karşılayacak şekilde optimize edilir. Sentez editörü olarak FPGA üreticilerinin kendi yazılımları kullanılabildiği gibi Mentor Graphics Precision Synthesis, Synplicity Synplify, Syplify Pro veya Synopsis Design Compiler gibi diğer sentez editörleri de kullanılabilir. Sentez işlemi tamamlandıktan sonra “post synthesis netlist” olarak isimlendirilen bir dosya üretilir. Sentez işlemi için hangi yazılım kullanılırsa kullanılsın, üretilen bu dosyanın FPGA yongası içerisinde hangi bağlantılar ile oluşturulacağını belirlemek için FPGA üreticisinin kendi programının kullanılması gerekmektedir. Quartus II Fitter yazılımı, post synthesis netlist dosyası içerisindeki lojik ifadeleri seçerek hedef donanım içerisinde lojik yapılara dönüştürür ve FPGA içerisindeki mevcut bağlantı yollarını kullanarak bu yapıları birbirlerine bağlar. Quartus II Fitter yazılımı, tasarımcının belirlediği veya varsayılan alan, performans, zaman ve güç kısıtlarına göre “post fit netlist” isminde bir dosya üretir. Bu dosya daha sonra bir dizi işlemle daha geçirilir. İlk olarak Quartus II TimeQuest isimli yazılım ile oluşturulan donanımın modelinin zaman kriterlerini karşılayıp karşılamadığı test edilir. Daha sonra, arzu edilirse bu donanım modeline kapı seviyesinde benzetim uygulanır. Bu benzetim RTL benzetimden farklı olarak bağlantı yollarından kaynaklanan gecikmeleri de dikkate alır. Dolayısıyla FPGA içerisinde oluşturulacak olan donanımın, FPGA yongasına yüklendiği zaman doğru bir şekilde çalışıp çalışmadığı bu sayede test edilebilir. Artık “post fit netlist” dosyasından FPGA'nın programlanmasına yarayan

konfigürasyon dosyası üretilebilir. Bu dosya JTAG ara yüzü ile SRAM hafızaya yüklenir. Bu işlemin ardından arzu edilirse Quartus II SignalTap programı ile FPGA'nın içerisine gömülmüş olan donanımın fiziksel ortamda doğru çalışıp çalışmadığı da kontrol edilebilir. Şekil 6.7'de bütün bu süreç özetlemiştir [119, 126].



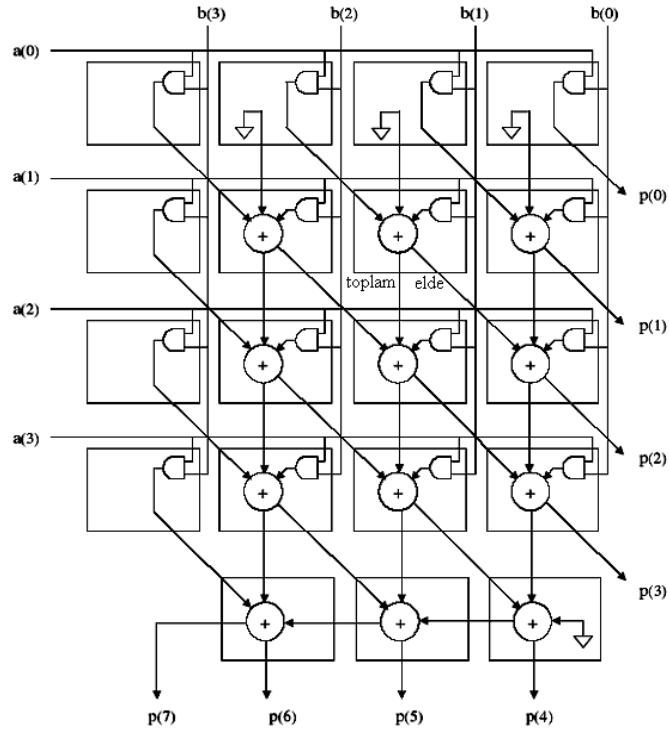
Şekil 6.7. FPGA ile tasarım süreci.

Sinyal işleme denildiğinde gerek akademi çevreleri gerekse ileri teknoloji kuruluşlarının sıklıkla başvurduğu bir araç olan Matlab, DSP ve FPGA tasarımcıları için de geliştirme araçları sunmaktadır. Matlab Simulink blok tabanlı tasarımda oldukça kullanışlı bir geliştirme platformudur. Simulink gerek Texas Instruments DSP yongaları için gerekse Altera ve Xilinx FPGA'ları için DSP tasarım geliştirme ortamı sunmaktadır. Şekil 6.8'de, Aletra DSP Builder ile Simulink ortamında modellenmiş bir filtre tasarımı görülmektedir. Altera DSP Builder programı bu model dosyasından HDL kod dizini oluşturarak, tasarımın FPGA üzerinde donanım olarak gerçekleştirilmesine olanak sağlar. Altera DSP Builder ile Simulink'in zengin kütüphanesi içerisindeki algoritma ve modeller, FPGA ile tasarım yapan uygulamacıların kullanımına sunulmuştur. İşlemsel doğruluk ve optimum algoritmayı belirlemede Matlab ve Simulink kütüphanelerinin sunduğu çözümlerin HDL kodlarına dönüştürülerek donanımlara aktarılması çok önemli bir avantajdır [127-129].



Şekil 6.8. Altera DSP Builder ile Simulink ortamında tasarlanmış bir model.

FPGA, tasarımcılara işlemcilerin sahip olmadığı bir dizi avantaj sağlar. Bunların belki de en önemlilerinden biri, paralel çalışan çok hızlı donanımların FPGA içerisinde oluşturulabilmesidir. Çünkü FPGA mimarisi, sıralı işlem birimlerinden farklı olarak paralel işlem yeteneğine sahiptir. Çok adımda yapılacak bazı işlemler, FPGA içerisinde oluşturulacak uygun bir lojik devre ile tek kenar tetiklemesinde yapılabilmektedir. Şekil 6.9’da çarpma işlemini sadece kapıların yayılım gecikmesi kadar bir sürede tamamlayabilen bir paralel çarpıcı devresi görülmektedir. Burada a ve b girişlerine 4 bitlik çarpım terimleri paralel olarak uygulanır uygulanmaz cevap üretilir. Bu tip çalışmaya “paralel çalışma” adı verilir ve işlem süresi kapı elemanlarındaki toplam yayılım gecikmesi kadardır. Bu yapı, tasarımcı tarafından HDL kodları veya hiç kod yazmadan sadece şematik editör ile tasarlanabilir. Çok sayıda HDL programlama dili vardır, bunlardan en çok kullanılanları Çok Hızlı Entegre Devre Donanım Tanımlama Dili (VHDL, Very High Speed Integrated Circuit HDL) ve Verilog’dur [120, 124, 130].



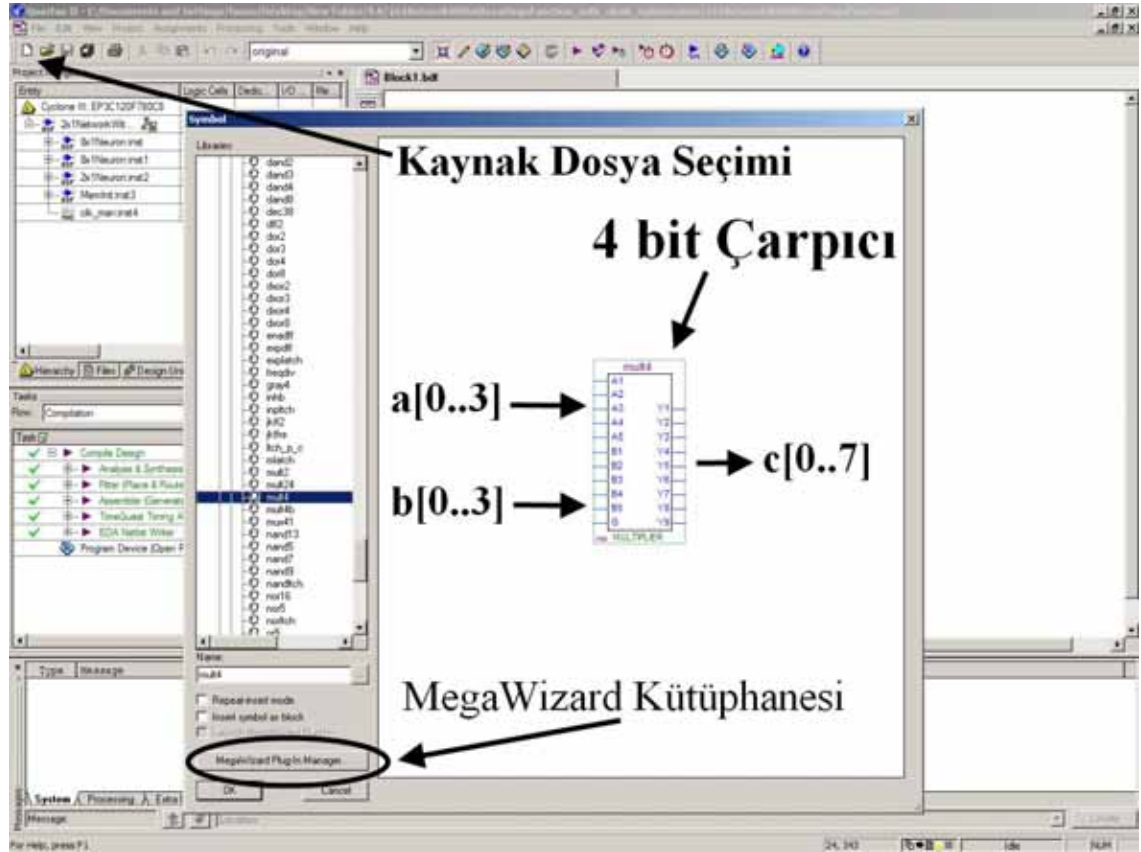
Şekil 6.9. Dört bit paralel çarpıcı devresi.

FPGA tasarım editörleri tasarımcıya farklı esneklikler sağlamaktadır. Örneğin Şekil 6.9'da gösterilen 4 bitlik paralel çarpıcı devresi, Altera Quartus II FPGA editörünün kütüphanesi içerisinde önceden tanımlanmış bir elemandır. Bu yapı aşağıdaki gibi bir VHDL kod dizini oluşturulabileceği gibi, Şekil 6.10'da verilen şematik editörden sembol şeklinde de kullanılabilir.

```

1  -----
2  library ieee;
3  use ieee.std_logic_1164.all;
4  use ieee.std_logic_arith.all;
5  -----
6  entity multiplier4 is
7  port ( a, b: in signed (3 downto 0);
8  prod: out signed (7 downto 0));
9  end multiplier4;
10 -----
11 architecture behavior of multiplier4 is
12 begin
13   prod <= a * b;
14 end behavior;
15 -----

```



Şekil 6.10. Dört bit paralel çarpıcının şematik sembolü.

Gerek Simulink gerekse FPGA tasarım editörlerinin sunmuş olduğu güvenilir modellere ulaşmak ve bunları tasarımlarda kullanmak çok kolaydır. Şekil 6.10'da daha gelişmiş modellerin bulunduğu MegaWizard kütüphanesine erişim ikonu gösterilmiştir. MegaWizard kütüphanesi ile fourier dönüştürücü veya filtre gibi daha kompleks algoritmaları yerine getiren modellere ulaşmak ve bunları tasarımlar içerisinde kullanmak mümkündür. Ayrıca IP Core isimli başka bir kütüphanede sayısal işaret işlemeye ait daha birçok farklı algoritmanın da model dosyalarına ulaşmak mümkündür. Bazıları ücret karşılığında satın alınan bu modelleri tasarımcı kendi projelerinde kullanabilir. Ayrıca Altera Quartus II editörü SoPC Builder isimli Altera'nın başka bir yazılımı ile birlikte çalışmakta ve FPGA içerisinde kullanıcıya özel işlemciler tasarlanmasına müsaade etmektedir. İstenilen sayıda dış birim (giriş çıkış portları, VGA sürücü, i2c, serial, ethernet, usb vs.) ile uygulamaya özel işlemcilerin tasarlanması ve bu tasarımların FPGA içerisinde kullanılması mümkündür [131].

6.1.1. FPGA ile Tasarımın Avantajları

Bugün tasarım şirketlerinde çalışan birçok mühendis, kısa zaman içerisinde sınırlı kaynaklar ile yeni şeyler tasarlarken, işverenler onlardan yüksek performans bekler ve ürettikleri şeyin farklılıklarla pazarda söz sahibi olmasını isterler. Ayrıca çok hızlı değişen teknoloji müşterilerin isteklerinin de farklılaşmasına sebep olmaktadır. Bir mühendis tasarımında yeni bir teknolojiyi kullanmayı düşündüğünde, kullanacağı teknolojinin mevcut en iyi olup olmadığını ve bu yatırımın iyi bir geri dönüşü olup olmayacağını düşünmek zorundadır. Bu zorunluluk günümüz tasarım mühendislerinde ciddi anlamda strese neden olmaktadır. 20 yıldan uzun bir süredir, FPGA gibi programlanabilir lojik devreler bu sıkıntıların giderilmesi noktasında pek çok katkıda bulunmuştur. FPGA devreler teknolojiye liderlik ederek esnek, uyarlanabilir ve düşük maliyetli çözümler ile ürünün pazara çıkma süresini kısaltmaktadır. Sahada çalışan bir FPGA yongasına dahi uzaktan erişilerek donanım güncellemesi yapılabilir. Bu işlem tasarımcının üzerindeki son ürün stresini çok önemli ölçüde azaltan bir avantajdır. Ayrıca Quartus II gibi FPGA tasarım geliştirme platformları takım çalışması için uyumlu yazılımlardır. Mühendisler projenin değişik birimleri üzerinde çalışabilir ve parçalar ortak platformda birleştirilerek proje tamamlanabilir.

6.1.2. Kullanım Alanları

FPGA ile tasarımda uygun donanımların kütüphanelerden hazır olarak kullanılması mümkündür. Aynı sayıda çarpım terimini içeren bir çarpma işlemi, mikroişlemci üzerinde yapıldığında çarpım sonucunun elde edilmesi için daha fazla sayıda saat darbesine ihtiyaç duyacaktır. Bazı DSP ve ASIC yongalarda bu tür işlemlerin minimum sürelerde yerine getirilmesi için özelleşmiş çarpıcı birimler kullanılmaktadır. Ancak işlemcilerde çarpım işlemi sayısı birden fazla olduğu durumlarda, bu özelleşmiş hızlı çarpıcının paylaşılması yani ikinci çarpma işleminin yapılabilmesi için birinci çarpma işleminin tamamlanması beklenecektir. Dolayısıyla paralel işlemin sunduğu avantajlar DSP yongalarla sağlanamamaktadır. Bununla birlikte FPGA, uyarlanabilir donanım yapısı, işlemsel kabiliyeti, tasarım süreçlerindeki hızı, düşük üretim maliyetleri ve FPGA tasarım editörünün sunduğu kolaylıklarla tasarımcıların ilgi odağı olmaktadır. FPGA sadece temel aritmetik işlemleri uygulayan mimarilerde değil sayısal tasarımın ihtiyaç duyulduğu pek çok uygulamada da kullanılmış tümleşik bir tasarım geliştirme platformudur [124, 130, 132-135].

ASIC yongaların tasarım ve üretim süreçleri çok uzun ve maliyetleri çok yüksek olduğu için bir ürünün ortaya konulması kolay değildir. Bu sebeple ASIC yongalarda ilk örnek tasarımlar sıklıkla FPGA'lar üzerinde gerçekleştirilmektedir. Bütün sayısal tasarımları FPGA üzerinde gerçekleştirmek mümkündür. Analog yapılar ayrı bir ön veya sonlandırıcı devre ile modellenenir. Analog ön devreli sayısallaştırılmış sistemlerde yüksek hızda doğru verilerin elde edilmesi için birçok yapı kullanılabilir. FPGA yüksek hız, esnek donanım, programlanabilir yapısı ve düşük güç tüketimi ile devre gerçeklemelerinde önemli bir avantaj sunmaktadır [128].

ASIC yongalar üretildikten sonra üzerlerinde bir değişiklik yapmak mümkün değildir. İşlemci tabanlı sistemlerde ise algoritmaları değiştirmek mümkün iken, iç donanımlar sabittir. Ayrıca, PCB üzerine dizilmiş olan bir sistemde donanımsal değişiklikler yapmak hem zaman alıcı hem de külfetlidir. Yapılacak her türlü donanımsal değişiklik yeni bir PCB ile gerçekleştirilir.

İşlemci tabanlı tasarımlar yapan mühendisler bir sistemi tasarlarlarken, ihtiyaçları ve bu ihtiyaçları karşılayacak çözümleri dikkatle tanımlayarak, arzu edilen özelliklerde bir mikroişlemci belirler. Fakat genelde arzu edilen özellikleri birebir karşılayan bir model bulunamaz. Mevcut yapılar ya ihtiyacın altında ya da ihtiyacın üstündedir. Fakat FPGA teknolojisi ile istenilen özellikte soft-core mikroişlemci tasarlamak mümkündür. FPGA üzerinde analog yapılar oluşturulamamakla birlikte ön ve sonlandırıcı analog/sayısal dönüştürücüler ile karma bir sistem işler hale getirilebilmektedir. Fakat bu, analog yapıların programlanabilir elemanlar ile oluşturulamayacağı anlamına gelmemektedir. Alan Programlanabilir Analog Diziler (FPAA, Field Programmable Analog Arrays) bazı analog yapıların yonga üzerinde gerçeklenmesine imkân tanımaktadır. FPGA ve FPAA yongaları birlikte kullanıldığı uygulamalar da bulunmaktadır [136, 137].

6.2. YSA'nın Sayısal Sistem Modellemesi

Biyolojik beyin benzetimi olan YSA matematiksel bir modele sahiptir. Bilgisayarlar üzerinde benzetimleri yapılabildiği gibi devre gerçeklemeleri de mevcuttur. YSA'nın doğasındaki paralellik, gerçek anlamda sadece donanımsal olarak oluşturulabilmektedir. YSA benzetim uygulamaları için Neumann tipi bilgisayar mimarileri sıklıkla tercih edilmektedir. Fakat bu tip benzetimlerde sinir ağının boyutu arttıkça benzetim zamanı da artmaktadır. Dahası sıralı işlem mantığı YSA'nın doğasına aykırıdır.

YSA'ların etkinliklerini arttırmak ve sahada kullanımını mümkün kılmak için bilgisayardan bağımsız donanımsal modellerinin oluşturulması üzerine çalışmalar yapılmaktadır. Bu çalışmalar nöro donanımlar (neuro hardware) başlığı altında incelenebilir. Nöro donanımlar analog, sayısal ve karma yöntemlerle, VLSI veya ASIC yongalar üzerinde oluşturulmuştur [37]. Analog YSA gerçeklemeleri düşük maliyet ve yüksek hız gibi avantajlarına rağmen zayıf hata bağışıklıkları ve sabit mimarileri büyük birer dezavantajdır. Diğer taraftan sayısal YSA gerçeklemeleri ise çok fazla donanım kaynağı kullanan büyük ölçekli çarpıcılar ve NAF bloklarına ihtiyaç duyarlar. Bu sebeplerden dolayı genellikle YSA'lar, tek bir çarpıcı ve NAF bloğunun paylaşarak kullanıldığı güçlü mikroişlemciler veya DSP'ler üzerinde gerçekleştirilirler. Fakat mikroişlemciler ve DSP'lerin sıralı işlem özelliği, YSA'nın paralel mimarisinin gerçekleştirilmesi için uygun bir geliştirme ortamı sunamaz. VLSI ve ASIC seçenekleri ise gerek çok uzun tasarım süreleri ve çok pahalı ilk üretim maliyetleri, gerekse sundukları sınırlı esneklikleri gibi dezavantajları yüzünden YSA donanım gerçeklemeleri için çok uygun değildir. FPGA gerçeklemeleri ise tasarımcıya ASIC ve VLSI gerçeklemelerine göre daha düşük bir maliyet, hızlı ilk örnekleme, uyarlanabilir donanım, güçlü yazılım geliştirme araçları desteği ve dağıtılmış paralel mimari gibi avantajlar sunar [38].

FPGA tabanlı YSA gerçeklemelerinde en önemli kıstas, mimarinin nasıl etkin bir şekilde kullanılacağıdır. İşlemci elemanlarda öncelikle girişler ilgili ağırlıklarla çarpılır ve daha sonra da bu çarpımlar toplanarak aktivasyon fonksiyonuna uygulanacak sinaptik çıkış üretilir. Çarpma işlemi yüzünden ihtiyaç duyulan alanın artması, FPGA tabanlı YSA'nın boyutunu belirleyen en önemli etkenlerden birisidir. Stanphen Bade tarafından geliştirilen yetenekli bir look-up-table (LUT) yöntemi ile önemli bazı başarımlar sağlanmıştır [138]. . Bu yöntem yüksek seviyede yoğunlaştırılmış bir işlemci eleman oluşturmak için, doğrusal olmayan aktivasyon fonksiyonunun davranış karakteristiğini bir hafıza elemanı ile gerçekleştirmektedir. Ayrıca H. Hikawa [139] işlemci elemanlar için aktivasyon fonksiyonu olarak üç değerli bir fonksiyon önermiştir. Fakat bu mimariler bit hassasiyetindedirler ve yüksek hassasiyet sağlayan 32 bit kayan noktalı nümerik tanımlamalar ile kullanılamazlar ve düşük bitlerle ifade edilen düşük doğruluklu sayı formatları ile kullanılabilirler. Bu sınırlama ile çarpma işlemleri bit dizileri çarpımı şeklinde standart ikili sayı aritmetiği ile yapılmaktadır.

Analog ve sayısal gerçeklemelerdeki karmaşıklıkları gidermek amacıyla araştırmacılar darbe modlu (pulse-mode) mimari kullanarak YSA donanımı geliştirme üzerine çalışmaktadırlar. Darbe modlu mimarinin avantajları, darbe modülasyon teknikleri ile birleştirilmiş temel sayısal devreleri kullanan sinaptik çarpıcı yapının, frekans modülasyon tekniklerini kullanması durumunda Direkt Dijital Frekans Sentezleyici (DDFS) ile çarpım sonucu üretilebilmesi ve doğrusal olmayan aktivasyon fonksiyonlarının en basit gerçekleştirilmesi olmasıdır. Dezavantajı ise sinaptik ağırlıkların -1.0 ile +1.0 arasında sınırlı bir aralıkta değişmesidir. Küçük farklar içeren sinyalleri sınıflandırmak için eğitilmiş çok katmanlı bir sinir ağının, bu küçük farklılıkları kuvvetlendirmek için büyük ağırlıklara sahip olması gerekmektedir. Bu yüzden küçük ağırlık aralığı büyük bir problemdir. Hikawa bu yapıyı geliştirerek parçalı doğrusal aktivasyon fonksiyonlu bir darbe modlu işlemci eleman önermiştir [140]. Önerilen yapıda aktivasyon fonksiyonu giriş sayısından etkilenmemektedir. Ağırlık değerlerinde de herhangi bir sınırlandırma söz konusu değildir. Fakat bu yapı sadece sabit noktalı tanımlamayla binary formatta oluşturulabilmektedir.

Literatürde yapılan çalışmaların yanında piyasada da uygulama türüne veya kullanılan nümerik tanımlama tipine göre değişik seçeneklerde üretilmiş ticari nöro donanımlar bulunmaktadır. Nöro donanımların etkili bir şekilde kullanılabilmesi için tasarım tipine uygun bir sınıflandırma probleminde kullanılması gerekir. Nöro donanımlar sistem mimarisi, paralellik derecesi, çoklu işlemci kullanılıyor ise her bir işlemcinin sorumlu olduğu ağ bileşeni, haberleşme şekli ve nümerik tanımlama tipine göre sınıflandırılabilir. Nöro donanımlar bağımsız işlem birimleri halinde çalışabildikleri gibi bir bilgisayar ile birlikte hızlandırıcı donanım olarak kullanılan şekilleri de mevcuttur [141]. Piyasadaki bazı ticari nöro donanımlar Tablo 6.1’de verilmiştir.

Her ne kadar Neumann tipi işlemcilerde çok önemli performans iyileştirmeleri sağlansa da sıralı işlem, gerçek zamanlı cevabı ve ağırlık eğitimini uzatmaktadır. Nöro donanımlar sahip oldukları paralel işlemci elemanlar vasıtası ile cevap süresi ve ağırlık eğitimini kısaltmaktadır. Çok hızlı işlem gücünün gerekli olmadığı uygulamalarda YSA modelleri genellikle bilgisayar üzerinde yazılımsal olarak çözümlenmektedir. Nöro donanımlar ASIC olarak sayısal veya analog yapılarda, sabit veya kayan noktalı nümerik tanımlamalar kullanılarak yapılmıştır. Nümerik hassasiyetin önemli olduğu uygulamalarda kayan noktalı nümerik tanımlamalar kullanılırken, hızın önemli olduğu

uygulamalarda ise sabit noktalı nümerik tanımlamalar kullanılır. Sabit noktalı nümerik sayı formatı hızı arttırdığı gibi maliyetleri önemli ölçüde azaltmaktadır. Bununla birlikte nümerik hassasiyette de önemli kayıplara sebep olmaktadır.

Tablo 6.1. Marketteki nöro donanımlar [126].

Nöro-donanım	Tipi	Algoritma	Performans	İşlemci Eleman
CNAPS	Nöro-Chip	Geri Yayılım SOM	1.6 GCPS	64
SYNAPSE-1	Nöro-Chip	Geri Yayılım Hopfield Ağı	33 MCPS	
Hitachi WSI	Nöro-Chip	Geri Yayılım	150 MCPS	144
Ni 1000	Hızlandırıcı Donanım	Geri Yayılım RBF Ağı	8.2 GCPS	
NT 6000	Hızlandırıcı Donanım	Geri Yayılım Kohonen Ağı	2 MCPS	
IBM ZISC	Hızlandırıcı	RBF Ağı		36

Nöro donanımlarda bir performans ölçüsü olarak Saniyede Hesaplanan Bağlantı (CPS, Connections Per Second) değerleri kullanılır. CNAPS (Connected Network of Adaptive Processor) The Adaptive Solution firması tarafından üretilmiş bir nöro bilgisayardır. Kaskat olarak bir veri yolu üzerinden birbirlerine bağlanmış 64 adet işlemsel elemandan oluşan N6400 nöro-chip yongası, içlerinde geri yayılım veya SOM algoritmalarının da bulunduğu birçok modelin gerçekleştirilebildiği bir donanımdır. CNAPS içerisinde yapay sinir ağının ağırlık değerlerinin saklandığı bir de hafıza elemanı bulunmaktadır. CNAPS ilave N6400 yongaları ile ağın büyütülmesine de imkân tanımaktadır. Geri yayılım algoritması ile 8 veya 16 bit ağırlık değerleri için 1.6 GCPS hızına sahiptir [126, 141, 142].

SYNAPSE 1 (Sythesis of Neural Algorithms on Parallel Systolic Engine) Siemens'in MA-16 nöro-chip yongası ile çalışan bir nöro bilgisayardır. Motorola MC68040 işlemcisi tarafından kontrol edilen 8 adet MA-16 yongasından oluşan bu donanım, aktivasyon fonksiyonu cevabını LUT yöntemi ile gerçekleştirmektedir. İçlerinde geri yayılım ve Hopfield algoritmalarının da bulunduğu birçok algoritmayı 16 bit sabit noktalı nümerik tanımlama ile gerçekleştirir [126, 141, 143].

Hitachi WSI 144 adet işlemci elemandan oluşan, 16 bit uzunluklu ağırlık değerlerine ve 9 bit uzunluklu aktivasyon fonksiyonu tanımlamalarına sahip olan bir nöro bilgisayardır [126, 144].

Nöro bilgisayarların yanında, daha ucuz maliyetli ve gelişmiş yazılım desteğine sahip olan ve bilgisayarların ana kartları üzerindeki ISA veya PCI genişleme yuvalarına monte edilen hızlandırıcı nöro donanımlar da bulunmaktadır. Bu donanımlardan birkaçı aşağıda verilmiştir.

Ni 1000, Nester/Intel tarafından üretilmiş ve örüntü sınıflandırmada kullanılan çok hızlı bir hızlandırıcı donanımdır. 16 bit bir mikroişlemci ve bilgisayar iletişim hattı bulunan bu hızlandırıcı donanım üzerinde, geri yayılım ve RBF algoritmaları koşturulabilmektedir. Her biri 5 bit uzunluklu 256 adet özellik vektörünü, 33 sınıfa ayırabilen bu donanım saniyede 33.000 örüntüyü sınıflandırabilmektedir [126, 145].

NT 6000 geri yayılım ve Kohonen algoritmalarının koşturulabildiği bir hızlandırıcı donanımdır. Bu donanım, eğitim algoritmasının koşturulması ve giriş verilerinin uygun bir şekilde donanıma beslenmesi için genellikle bir DSP işlemci ile birlikte kullanılmaktadır. Hızı tipik olarak 2 MCPS değerindedir [126, 144].

ZISC (Zero Instruction Set Computer) IBM firması tarafından üretilmiş bir nöro-chip yongasıdır ve özel bir eğitim algoritması ile eğitilen RBF tipi ağların, 36 işlemci eleman ile oluşturulmasına imkân tanımaktadır. 16 ZISC yongasından oluşan ve toplam 576 işlemci elemana sahip, ISA uyumlu bir hızlandırıcı kart modeli de mevcuttur. 64 adet işlemci eleman bulunan bir hızlandırıcı, 8 bit uzunluklu giriş verisi için saniyede 165.000 örüntüyü sınıflandırabilir [126, 144].

Geçmişte araştırmacılar YSA'nın büyüklüğünün performansa etki eden en büyük parametre olduğunu düşünürlerdi. 1980'lerde nöro bilgisayar tasarımlarında temel mantık çok sayıda genel amaçlı işlemcinin paralel bağlanarak, büyük bir ağı hesaplama yükünün paylaştırılmasıyla hızlı bir şekilde işlemlerin yapılmasının sağlanmasıydı. 1990'larda CAM-Brain projesi ile YSA zekâsının sadece ağı boyutu ile tanımlanamayacağı ortaya konulmuştur [146]. Bugün YSA araştırmacıları tarafından ortaya konulan genel fikir, YSA başarımında ağ mimarisi ve iç bağlantıların önemli birer parametre olduğudur.

Ticari nöro donanımlar ASIC mimariye sahip olduklarından ve genel anlamda gerçek zamanlı uygulamalara çözüm olarak geliştirildiklerinden, çok özel uygulamaların dışında yaygın bir kullanıma sahip olamadılar. Ayrıca gerek maliyet, gerek temin edilebilirlik açılarından da pratik uygulamalarda bir avantaj oluşturmamışlar ve çoğunlukla bilimsel araştırma projelerinde sınırlı bir uygulama alanı bulabildiler. FPGA teknolojisinin gelişmesi ile nöro donanımlar uygulamaya özel olarak, FPGA yongaları üzerinde tasarlanmaya başlanmıştır. FPGA donanımları gerçek zamanlı uygulamalar için gereken hesaplama hızını, esnek mimarileri ile daha ucuz donanımlarla oluşturmayı mümkün kılmıştır [21, 147].

6.3. Nümerik Tanımlama

FPGA üzerinde YSA geliştirme uygulamalarında donanım maliyetini etkileyen bir diğer önemli parametre ise tasarımın kullandığı nümerik tanımlama yöntemidir. Aşağıda sayısal hesaplamalarda kullanılan günümüz modern nümerik tanımlama metotlarının oluşması süreci ve bu tez çalışmasında kullanılan iki farklı sayı sistemi hakkında bilgiler verilmiştir.

Rakamların insanlık tarihinde kullanılması 4000 yıllık bir geçmişe sahiptir. Günlük hayatımızda kullandığımız desimal (onluk) sayı sistemi, MS 500 yıllarında insan el ve ayaklarında bulunan 10 parmağın esinlenilerek Hindular tarafından bulunmuştur [148]. Roma rakamlarının matematiksel yorum zorluğu ve işlevsel zayıflıkları, İlk defa Hindu sayı sisteminde bir sayı olarak ifade edilen “sıfırın” tanımı ile aşılmıştır. Dokuzuncu yüzyılda Araplar tarafından geliştirilen bu sistem, sıfırın avantajını kullanmak amacıyla ancak 16. yüzyılda Avrupalılar tarafından kullanılmaya başlanmıştır [149]. Avrupa’da Arap sayıları olarak ifade edilen desimal (ondalık) sayılar, Araplar tarafından Hindu sayıları olarak ifade edilir. Bu iki kültürü de tanımak amacıyla, desimal sayılar genellikle Hindu-Arap sayıları olarak ifade edilir [148, 150, 151].

Günümüz bilgisayar mimarilerinde kullanılan modern hesaplama sisteminin temeli olan ikili (binary) sayı sistemi, ilk defa dördüncü yüzyılda şiirlerindeki uzun heceleri matematiksel biçimde kısaltmak amacı ile Hindu şair ve yazar Pingala tarafından geliştirilmiştir [151]. İkili sayı sistemi 1500 yıl aradan sonra ilk defa, 1600’lü yıllarda deneyciliğin babası olarak bilinen İngiliz bilim adamı Francis Bacon tarafından, ikili kodlama (binary encoding) teorisi ile yeniden yorumlanmıştır. 1854 yılında İngiliz

matematikçi George Boole, Boole Cebri (Boolean Algebra) olarak bilinen ikili aritmetiği, çalışması ile tanımlamıştır. 1937 yılında bilişim teorisinin babası olarak bilinen Amerikalı elektronik mühendisi Claude Shannon, elektronik devrelerde kullanılan kapı (röle) sayılarının, ikili sayı sistemi ve Boole Cebri kullanılarak nasıl azaltılacağını tarif etmiş ve bu çalışması ile sayısal bilgisayar ve sayısal devrelerin temellerini atmıştır.

Sayısal sistemlerin ortaya çıkması ile bu mimarilerde kullanılmak üzere, sayısal hesaplama teknikleri ve nümerik tanımlamalar ortaya çıkmıştır. Modern bilgisayar mimarilerinde kullanılan sayı formatları ve tarihsel gelişim süreci aşağıda kısaca özetlenmiştir.

En eski bilgisayardan üçü ENIAC [152], UNIVAC [153] ve IBM 650 [154] aritmetik işlemleri desimal olarak yani onluk sayı sisteminde gerçekleştirirken [155], yine aynı dönemde EDSAC [156], EDVAC [157] ve ORDVAC [158] bilgisayarları aritmetik işlemleri binary olarak yani ikili sistemde gerçekleştirmekteydiler. Bilgisayarların yapı taşı olan iki durumlu transistörlerin binary sayı sistemindeki avantajına rağmen, bazı bilgisayar üreticileri dört bit ikili kodlanmış onlu (BCD, Binary Coded Decimal) [159] sayı sistemini kullanmaya devam ettiler. Zamanla uygulamalarda ihtiyaç duyulan yüksek hızlı hesaplama, binary aritmetiğin kullanımını gerektirmiş ve binary aritmetiği bir hesaplama standardı haline gelmiştir [151].

Sabit noktalı nümerik tanımlamanın aksine, bilgisayar üreticileri kendi makineleri için kendilerine özgü kayan noktalı nümerik tanımlama standartları geliştirmişlerdir [160]. Bunun sonucu olarak farklı bilgisayarlarda kullanılan farklı kayan noktalı nümerik tanımlama algoritmaları yüzünden, aynı veri farklı sonuçlar ile yorumlanmaktaydı. Bu yüzden ortak bir kayan noktalı nümerik tanımlama standardına ihtiyaç duyulmuştur. 1985 yılında IEEE tarafından 754 numaralı referans ile tanımlanan standart ile kayan noktalı nümerik tanımlama yoğun bir şekilde kullanım alanı bulmuş [161] ve bugün revize edilmiş hali olan IEEE 754 2008 standardı ile son haline getirilmiştir [162]. Bütün bu süreç içerisinde bilgisayar programlama dilleri de sabit noktalı nümerik tanımlamadan, kayan noktalı nümerik tanımlamaya doğru evrimleşmiştir [151].

6.3.1. Sabit Noktalı Nümerik Tanımlama

Sabit noktalı nümerik tanımlama tamsayı (integer) sayı formatına benzemektedir. Ancak sabit noktalı nümerik tanımlamada, kesiri belirleyen radix noktası integer sayı formatında olduğu gibi sağdaki en düşük değerli bitin (LSB, Least Significant Bit) hemen sağında yer almayabilir. Radix noktasının yeri bellidir ve bu noktanın sağındaki solundaki bit sayıları sabittir. Sabit noktalı nümerik tanımlamada aritmetik işlemlerde kullanılan bütün operantlarda radix noktasının yeri aynıdır. Radix noktasının sağında kalan bitler fraction (F) olarak ifade edilir ve küsuratı ifade ederler. Bu noktanın solunda yer alan bitler ise integer (I) değeri ifade ederler. Sabit noktalı nümerik tanımlamada noktanın sağındaki ve solundaki bitler yani veri uzunluğu, hassasiyeti belirler. Sabit noktalı nümerik tanımlamada kullanılan bit sayısı, tanım aralığını ve hassasiyeti belirler. Fakat bit sayısı sınırlı olduğu için tasarımcı aritmetik işlemler sonucunda oluşacak nümerik kayıpları ve taşmaları göz önünde bulundurmalıdır.

Bu tez çalışmasında soldaki en yüksek değerlikli bit (MSB, Most Significant Bit) işaret biti, sonrasındaki yedi bit integer (tamsayı) ve kalan sekiz bit fraction (kesir) olmak üzere, 16 bit sabit noktalı nümerik sayı formatı kullanılmıştır. 64,125 değeri için binary ifade Şekil 6.11’de verilmiştir.

$\pm$	2^6	2^5	2^4	2^3	2^2	2^1	2^0	.	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}
0	1	0	0	0	0	0	0	.	0	0	1	0	0	0	0	0
1	2	3	4	5	6	7	8		9	10	11	12	13	14	15	16
MSB	I											F				LSB

Şekil 6.11. 16 bit sabit noktalı nümerik sayı formatı.

16 bit nümerik tanımlama ile yaklaşık ± 128 genişliğindeki nümerik aralık, $p = 2^{-8}$ nümerik hassasiyet ile ifade edilebilmektedir. Radix noktası sağ ve solundaki bitleri eşit olarak sekizer bite bölmüştür ve bütün hesaplamalarda bu noktanın yeri sabittir. Nokta bit dizisi içerisinde görüntülenmez, tasarımcı noktanın yerini bilerek işlemleri yapar. Bütün aritmetik işlemler temel binary aritmetiği mantığı ile gerçekleştirilir. Çarpım işlemleri sonucunda, sonuç bit dizisi iki katına çıkar. Yani 16x16 çarpım sonucunda 32 bitlik bir cevap oluşur. Bu değer tekrar 16 bit ile ifade edildiğinde veride taşma oluyorsa sonuç hatalı çıkacaktır. Bu tez çalışmasında kullanılan veri seti ve ağırlıklar taşmalara sebep olmayacak şekilde seçilmiştir.

VHDL kütüphanesi işaret bitli sabit noktalı nümerik sayı formatını desteklemektedir. MSB'den yani işaret bitinden sonraki bitlere binary aritmetiği uygulanır. İşlem sonucunda elde edilen 15 bitlik veri için işaret biti atanır ve sonuç tekrar 16 bit ile ifade edilir.

Sayısal sistemlerde en kısa işlem süreleri sabit noktalı nümerik tanımlama ile yakalanmaktadır. Sabit noktalı nümerik tanımlama binary aritmetiğini kullandığı için donanım seviyesine en yakın olan tanımlama tekniğidir. Bu yapısal sadeliği yüzünden donanım kaynağı ihtiyacı da oldukça düşüktür. Zaten sabit noktalı nümerik tanımlamanın hızlı cevap süresinde en belirleyici katkı, verinin donanım içerisinde maruz kaldığı propagasyonun en az olmasından ileri gelmektedir. Az sayıdaki donanım kaynağı, veri akışındaki gecikmeleri minimize eder.

6.3.2. Kayan Noktalı Nümerik Tanımlama

IEEE'nin 754 numaralı standardı ile ilk defa 1985 yılında ortaya konulmuş olan kayan noktalı nümerik tanımlama standardı (IEEE 754 Floating Point Numerical Description) [161], binary sayı sisteminin ticari olarak yeni bir formatta yorumlanmasını sağlamıştır. Bugün artık IEEE 754 2008 [162] ile son haline getirilmiş olan bu nümerik format tarih boyunca en çok kullanılan nümerik standart olmuştur.

Kayan noktalı nümerik tanımlama en temel şekli olan 1985'te yayınlanan ilk standarda göre dört ana formatta, temel ve genişletilmiş olmak üzere iki grup altında ifade edilir. Bu dört kayan noktalı nümerik tanımlama formatı aşağıdaki gibi ifade edilmektedir.

Temel

- 32 bit kayan noktalı nümerik tanımlama, tek hassasiyetli (single precision)
- 64 bit kayan noktalı nümerik tanımlama, çift hassasiyetli (double precision)

Genişletilmiş

- 43 veya daha fazla sayıda bit ile ifade edilen kayan noktalı nümerik tanımlama (single-extended precision)
- 79 veya daha fazla sayıda bit ile ifade edilen kayan noktalı nümerik tanımlama (double-extended precision)

IEEE 754 2008 ile 16 bit hassasiyetten (half precision) 128 bit hassasiyete (Quadruple precision) kadar ifade edilebilen kayan noktalı nümerik tanımlama formatları geliştirilmiştir. Bununla birlikte bu formatlar 32 ve 64 bit kayan noktalı nümerik formatlar kadar geniş bir kullanım alanı bulamamışlardır.

Kayan noktalı nümerik tanımlama standardıyla sayıların değişik seçeneklerle (en yakın değere, pozitif sonsuza doğru, negatif sonsuza doğru ve sıfır etrafında) yuvarlanması, pozitif ve negatif yönlü taşmalar, sonsuz ve işaret bitli sıfır gibi özel sayılar tanımlanmıştır. Toplama, çıkarma, çarpma ve bölme gibi aritmetik işlemler de bu standart içerisinde tanımlanmıştır. Kayan noktalı nümerik tanımlama VHDL kütüphanesi tarafından da desteklenmektedir.

Kayan noktalı nümerik sayı formatı üç ana bileşene bölünebilir. Bunlar işaret biti (ib), mantissa (sabit noktalı nümerik formattaki fraction yani kesire karşılık gelmektedir) ve exponent (tamsayı) bileşenleridir. 32 bit nümerik tanımlama için kullanılan sayı formatı Şekil 6.12’de verilmiştir. 0 ile 22 arasındaki toplam 23 bit mantissa’yı, 23 ile 30 arasındaki 8 bit exponent’i ve en soldaki bit işaret bitini temsil etmektedir. İşaret biti sabit noktalı nümerik tanımlama formatında olduğu gibi 0 ise pozitif, 1 ise negatif değer tanımlar.

ib		Exponent (8 bit)				Mantissa (23 bit)					
31	30		23	22		0					

Şekil 6.12. 32 bit kayan noktalı nümerik sayı formatı.

32 bit kayan noktalı nümerik sayı formatı $\pm 2^{-126}$ ile $2^{127} (2 - 2^{-23})$ aralığını yani desimal olarak yaklaşık $\pm 1,8 \times 10^{-38}$ ile $3,4 \times 10^{38}$ aralığını tanımlar. Binary radix noktasının yeri sabit değildir ve bu yüzden kayan noktalı tanımlama adını alır. 32 bit nümerik sayı formatı aritmetik işlemler için binary formata göre daha büyük hafıza alanları işgal eder ve işlemleri daha uzun sürelerde tamamlar. Fakat mantissa’dan görüleceği gibi çok yüksek bir hassasiyet ile cevap üretir. Birçok yazılım veya uygulama için nümerik veri hassasiyeti büyük önem taşımaktadır. Kayan noktalı nümerik tanımlama ile tasarımcıya istediği kadar hassas nümerik işlem olanağı sunulmaktadır.

6.3.3. Sabit ve Kayan Noktalı Nümerik Tanımlamaların Karşılaştırılması

Sabit noktalı nümerik tanımlama, kayan noktalı nümerik tanımlamaya göre sınırlı kabiliyetlere sahiptir. Buna rağmen geçmişteki yonga teknolojilerinin sahip olduğu sınırlı tümleşik yoğunluktan dolayı sayısal sistemlerde en uzun süre kullanılmış olan sayı formatıdır. Bununla birlikte bazı yeni nesil teknolojiler, sonsuza yakınsayan büyük değerli veya sifıra yakınsayan küçük değerli sayılar kullanmaktadırlar. Fakat bazı uygulamalarda nümerik aralık kadar hassasiyet de önemlidir. Unutulmamalıdır ki 0 ile 1 arasında sonsuz sayıda nümerik değer bulunmaktadır. Kayan noktalı nümerik tanımlama çok büyük bir nümerik aralığı çok yüksek derecede hassasiyet ile ifade etmektedir.

Çok yüksek bir oranda doğruluk sağlaması sebebiyle, kayan noktalı nümerik format en ideal nümerik tanımlama yöntemidir. Bu özelliği sayesinde uygulamalarda en yaygın kullanılan çözümdür. Bu, neden kayan noktalı tanımlamanın genel amaçlı hesaplamalarda sıklıkla kullanıldığını ve birçok sayısal entegre üreticisinin özel kütüphaneler ile destek sunduğunu açıklamaktadır. Fakat FPGA yongalarındaki sınırlı kaynaklar sebebiyle nümerik tanımlamalar için kayan noktalı nümerik tanımlamanın kullanımı, sabit noktalı nümerik tanımlamada olduğu gibi daha etkili kaynak kullanan tanımlama yöntemlerinin kullanımı kadar uygun değildir [163].

Kayan noktalı nümerik tanımlamanın kullanıldığı donanımlar, sabit noktaliya ilave lojik kapılardan meydana gelir. 32 bit kayan noktalı çarpıcının kullandığı alan, 16 bit sabit-noktalı çarpıcının kullandığından çok daha fazladır. Aynı durum doğal olarak aktivasyon fonksiyonu, toplama ve çıkarma birimleri için de geçerlidir. FPGA tabanlı YSA’larda kayan noktalı formatın ihtiyaç duyduğu daha fazla kaynak ve alan ihtiyacı, ancak tasarlanan YSA boyutunun küçültülmesi ile FPGA gerçekleştirmesini mümkün kılar.

K.R. Nichols, M.A. Moussa ve S.M. Areibi FPGA tabanlı YSA yapılarını, kaynak kullanımları, optimum değerler vs. cinsinden test etmişlerdir [164]. BP öğrenme algoritması kullanılarak oluşturulan YSA ile XOR problemini aşağıda verilen mimariler ile çözmüşlerdir;

- i) 32 bit kayan noktalı nümerik tanımlama ile gerçekleştirilmiş FPGA tabanlı YSA,
- ii) 16 bit sabit noktalı nümerik tanımlama ile gerçekleştirilmiş FPGA tabanlı YSA,
- iii) Yazılım tabanlı YSA.

Klasik XOR probleminin çözümünde kullanılan bir YSA için mikroişlemci ve FPGA performansları Tablo 6.2’de görülmektedir. Buradan hassasiyetin, oluşan yoğunluk ve kullanılan kaynak sayısı üzerinde direk etkisi olduğu anlaşılır. Günümüz FPGA’ları göz önüne alınacak olursa düşük nümerik hassasiyete sahip olan 16 bit sabit noktalı nümerik tanımlama formatı en hızlı donanım cevabını üretmektedir.

Tablo 6.2. Mantıksal XOR problemini çözmekte kullanılan yapılar ve sonuçlar [164].

XOR YSA Mimarisi	Duyarlık (Hassasiyet)	Toplam Alan CLB [Slices]	Maksimum Clock (MHz)	Saniyede Değişen Ağırlık (WUPS)
Xilinx Virtex Excv2000e	16 bit Sabit Noktalı	1.239 [2.478]	10	62.762
Xilinx Viretx II Xc2v8000	32 bit Kayan Noktalı	8.334,75 [33.339]	1,25	5.172
Intel Pentium III	32 bit Kayan Noktalı	Sonsuz sayıda	733	1.467

Tablo 6.2’de görüldüğü gibi nümerik tanımlamada kullanılan bit sayısı azaldıkça donanım hızı artmaktadır ve kullanılan kaynak sayısı azalmaktadır. 10 MHz sistem clock girişi olan bir FPGA üzerinde 16 bit nümerik tanımlama ile oluşturulmuş YSA mimarisi saniyede 62.762 kez değişim gösterirken, aynı problemin çözümlendiği 300 MHz sistem clock girişi olan bir Pentium III işlemcide saniyede 1.467 değişim görünmektedir. 10 MHz clock girişi olan FPGA, 700 MHz ile çalışan güçlü bir işlemciden daha hızlı çalışmaktadır. Bunun sebebi FPGA’ların sahip olduğu paralel işlem yeteneğidir. FPGA üzerinde aynı yapı kayan noktalı tanımla kullanılarak tasarlandığında, değişimin 5.172’ye kadar düştüğü ve kullanılan kaynak sayısının ise arttığı görülmektedir. Bununla birlikte 32 bit nümerik tanımlamanın ağırlık nümerik hassasiyetini arttırdığı unutulmamalıdır. Bu bakımdan kaynak kullanımı ile nümerik tanımlama arasında bir ilişki vardır ve bir problemin çözümünde problemin ihtiyaç duyduğu kritere göre, doğruluk veya alan ölçütleri gözetilerek ağ tasarlanmalıdır.

6.4. 8x2x1 Boyutlu YSA'nın FPGA Üzerinde Gerçeklenmesi

Bu çalışmada donanım geliştirme platformu olarak Aletra Cyclone III EP3C120F780 yongası 50 MHz sistem clock frekansı ile kullanılmıştır. Cyclone FPGA ailesi gelişmiş fonksiyonlara sahip iç donanımları ile düşük güç tüketimi ve düşük maliyet avantajları sunar. EP3C120, Cyclone III ailesinin en büyük FPGA yongalarından biridir. Cyclone III ailesi düşük güç tüketimi ve maliyetine karşın yoğun bir donanım kaynağına sahiptir. Bu sayede tasarımcı büyük ölçekli devreleri tek bir FPGA yongası içerisinde oluşturabilir. Bu tez çalışmasında ailenin en büyük yongalarından biri olan EP3C120'nin kullanılmasının sebebi, donanım modeli oluşturulurken FPGA yongasından kaynaklı donanım sınırlandırmalarının yaşanmamasıdır. Son hali verilen bir sayısal tasarım, aynı aile içerisinde ihtiyacı karşılayabilecek daha küçük bir FPGA üzerinde kolayca gerçekleştirilebilir.

Paralel çalışan donanımlar genellikle giriş ve çıkışlarda her bir biti temsil edecek bir I/O terminali ve matematiksel ifadelerin ayrı ayrı fakat aynı zamanda koşturulduğu bağımsız matematiksel operatörlere ihtiyaç duyarlar. Cyclone III ailesine ait bazı yongaların sahip oldukları donanım kaynakları Tablo 6.3'te verilmiştir. EP3C120 18x18 bit genişlikli 288 adet gömülü çarpıcıya sahiptir, arzu edilirse gömülü çarpıcılar iki katı yani 576 adet 9x9 bit genişlikli çarpıcı olarak da kullanılabilir. Ayrıca gömülü çarpıcılara ilave olarak, EP3C120'nin sahip olduğu diğer donanım kaynaklarını kullanarak yapılandırılan, maksimum 16x16 bit genişlikli 432 adet yazılımsal çarpıcı daha oluşturulabilir. Tablodaki I/O sayısı maksimum değer olarak verilmiştir ve bu sayı EP3C120 için 531'dir. EP3C120 serisi içerisinde terminal sayıları ve yonga kılıf yapıları farklı olan FPGA yongaları bulunmakla birlikte, hepsi aynı sayıda donanım kaynağına sahiptir. Bu çalışmada kullanılan EP3C120 serisi FPGA yongasının genişletilmiş ismi Cyclone III EP3C120F780'dir. Sondaki F780 tanımlaması ise bu FPGA'nın kılıf tipini ifade eden bir belirteçtir. F780, 2.6 mm yüksekliğinde 29x29 mm boyutunda kare bir plastik kılıfı tanımlamaktadır. Bir FPGA yongasının sipariş kodunda bu genişletilmiş tanımlama ismin sonunda yer alan ve üç karakterden oluşan sırası ile çalışma sıcaklık aralığı, hız ve isteğe bağlı tanımlamaların bulunduğu bir son ek vardır. İsteğe bağlı tanımlamaların yapıldığı son karakter ise entegrenin ürün numunesi veya kurşunsuz olduğunu tanımlamak için verilebilir. Cyclone III Device Handbook isimli dokümanda bu ailenin bütün özellikleri detaylı olarak verilmiştir [119].

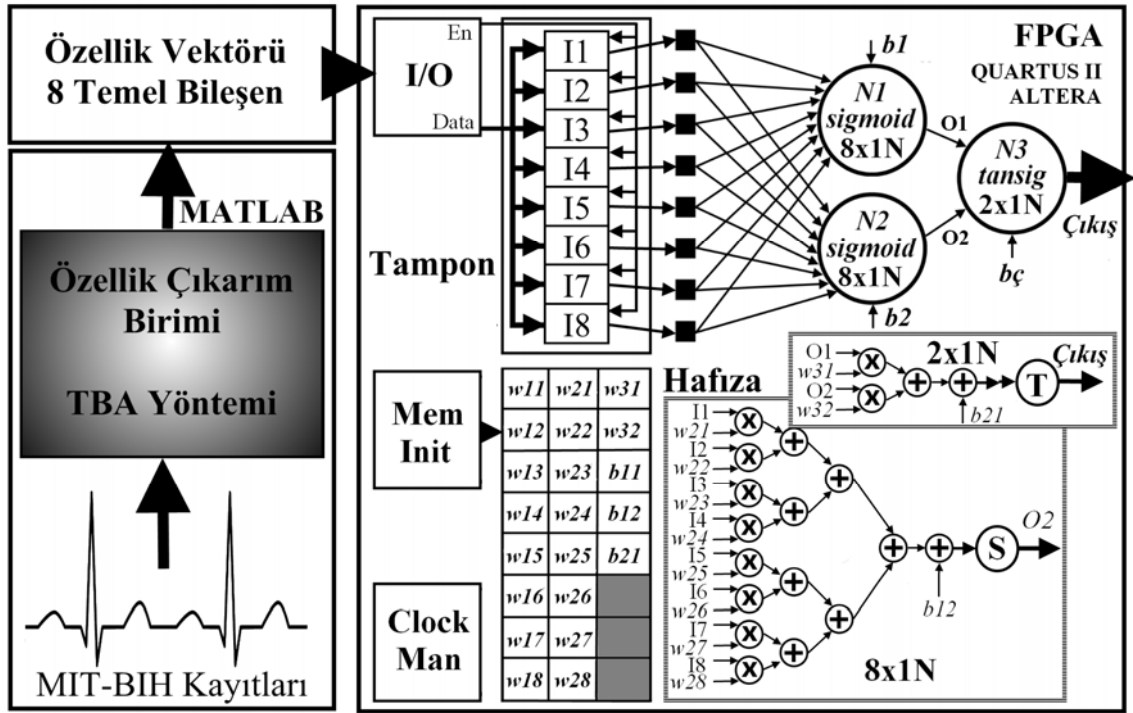
Tablo 6.3. Cyclone III ailesinden bazı FPGA'ların donanım kaynak özellikleri.

Aile	Yonga	Lojik Eleman	Toplam RAM (bit)	18x18 Çarpıcılar	PLL Sayısı	I/O Sayısı
Cyclone III	EP3C5	5.136	423.936	23	2	182
	EP3C25	24.624	608.256	66	4	215
	EP3C55	55.859	2.396.160	156	4	377
	EP3C80	81.264	2.810.880	244	4	429
	EP3C120	119.088	3.981.312	288	4	531

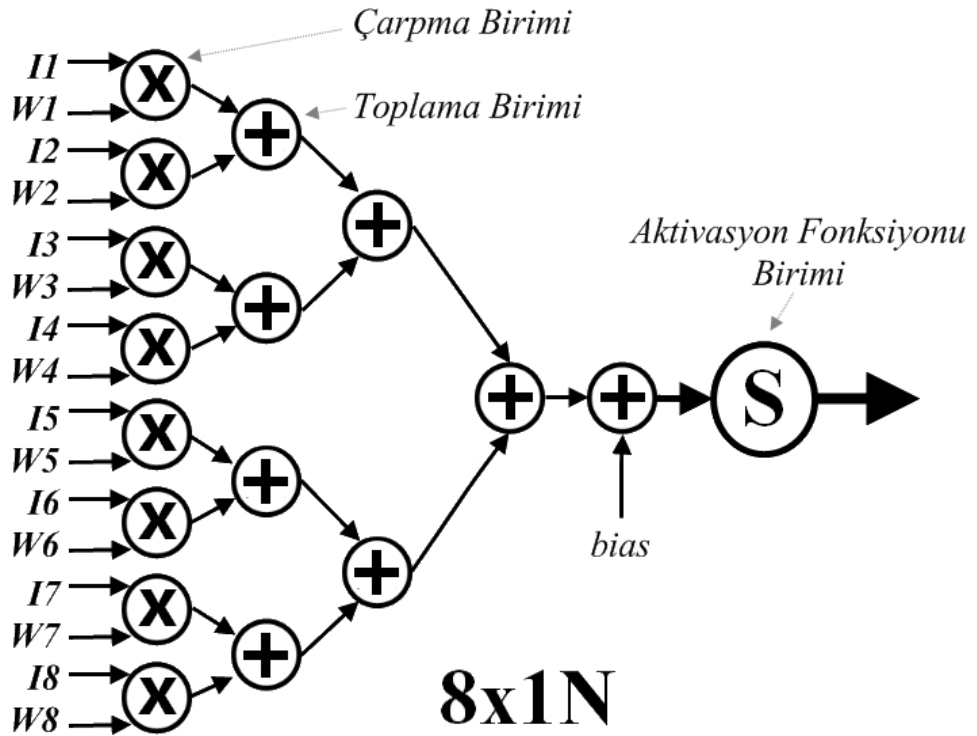
Cyclone III ailesi taşınabilir (mobil) uygulamalar için yüksek yoğunluklu hafıza, gömülü çarpıcı ve I/O seçenekleri sunar. Cyclone III birçok dış hafıza erişim ve I/O iletişim protokolünü destekler. Bu aile için özel fonksiyonları icra etmek üretilmiş çok sayıda uyarlanabilir Hazır Tasarım (IP, Intellectual Property) çekirdeği bulunmaktadır. Bu IP çekirdekleri birkaç küçük uyarlama ile tasarımların içerisine eklenerek kullanılabilir [119].

Bu tez çalışmasında Cyclone III EP3C120F780 FPGA yongası üzerinde, IEEE 754 32 bit kayan noktalı ve 16 bit sabit noktalı nümerik tanımlamalar kullanılarak, 8x2x1 boyutlu YSA tabanlı bir otomatik PVC aritmi sınıflandırıcı %97,66 ve %96,54 doğruluklarla tasarlanmıştır. Bu yapılar sırası ile FPANN ve FXANN olarak isimlendirilir. FPANN IEEE 754 32 bit kayan noktalı nümerik tanımlama kullanılarak FPGA üzerinde oluşturulmuş 8x2x1 boyutlu YSA modelinin donanım adıdır. FXANN ise aynı boyuttaki bu YSA modelinin 16 bit sabit noktalı nümerik tanımlama kullanılarak FPGA üzerinde donanım olarak tasarlanmış adıdır. FPANN donanımı gizli katmanında gerçek sigmoid aktivasyon fonksiyonu kullanılarak tasarlanmış, FXANN ise gizi katmanında parçalı doğrusal sigmoid aktivasyon fonksiyonu kullanılarak tasarlanmıştır. Hatırlanacağı üzere, FPANN ağ parametreleri 5. Bölüm'ün sonunda verilmişti, FXANN modelinin ağ parametreleri ise bu bölüm içerisinde, parçalı doğrusal sigmoid fonksiyonu ve bu fonksiyonun Matlab uyarlaması anlatıldıktan sonra verilecektir. 16 bit nümerik tanımlamanın ve parçalı doğrusal sigmoid aktivasyon fonksiyonunun kullanıldığı FXANN yapısı ağın hatasını arttırmış ancak donanım maliyeti ve cevap süresini azaltmıştır. Bu iki yapının kaynak kullanımları, donanım gerçeklemeleri ve performans ölçütleri bu bölümün sonunda detaylı olarak verilecektir.

FPANN ve FXANN birbirlerinden farklı özelliklere sahip donanımlar olmalarına karşın her ikisi de $8 \times 2 \times 1$ boyutunda olup fonksiyonel olarak özdeş birimlerden oluşmaktadırlar. $8 \times 2 \times 1$ boyutlu YSA'nın donanımsal olarak blok diyagramı ve donanımın Matlab ile bağlantısı Şekil 6.13'te verilmiştir. FPGA içerisindeki YSA donanımı şekilden görüldüğü gibi sekiz ana birim ve bunların çok sayıdaki alt birimlerinden meydana gelir. Bu birimleri kısaca açıklamak gerekirse, ClkMan birimi FPGA içerisindeki clock organizasyonunu sağlayan birimdir. Verinin yayılım yoluyla işlendiği paralel mimaride, her bir işlem birimi için bir önceki verinin hesaplaması bitene kadar pasif tutulur. Ayrıca işlemini tamamlayan her bir birimin çıkışı hafızaya alınarak, bu birimler de pasif duruma geçirilir. Ardışıl işlem bloklarının ne zaman ve ne kadar süreler ile çalışacakları ClkMan birimi tarafından kontrol edilir. Ancak bunun yapılabilmesi için hangi birimin ne sürede işlemini bitirdiğinin bilinmesi gerekmektedir. ClkMan birimi yanlış hesaplanmış verilerin oluşmasını önlemekte, sistem güç tüketimini düşürmekte ve de YSA'nın her girişini aynı süreler ile kontrol ettiği için verinin ağ üzerinde bütün girişler boyunca tümüyle paralel olarak yayılmasını sağlamaktadır. MemInit birimi Matlab'den elde edilen ağırlık ve biasların saklandığı gömülü hafıza birimidir. Eğitimi Matlab ortamında tamamlanan ağların, ağırlık ve bias değerleri bu hafızadan çağrılarak donanım cevapları oluşturulur. FPGA yongası ilk enerjilendiğinde, işlemci elemanlar kendi ağırlık ve bias değerleri ile güncellenir. I/O bloğu MLP YSA donanımının veri yoludur ve ağın girişidir. Tampon birimi veri yolu üzerinden aktarılan seri giriş bilgisinden sekiz girişi yakalayarak ayıklar ve bu girişler ile gizli katmandaki işlemci elemanları besler. 8×1 Nöron bloğu ($8 \times 1N$) tasarımın en büyük alan kaplayan birimidir ve içerisinde çok sayıda çarpıcı, toplayıcı gibi alt hesaplama birimleri barındırır. $8 \times 1N$ birimi içerisindeki sigmoid aktivasyon fonksiyonu birimi ise en çok kaynak tüketimine sahip olan alt birimdir ve Şekil 6.14'te blok diyagram olarak gösterilmiştir. 2×1 Nöron ($2 \times 1N$) birimi çıkış işlemci elemanıdır ve purelin aktivasyon fonksiyonuna sahiptir. FPANN için I/O birimi veri yolu 32 bit, FXANN için 16 bittir. Şekil 6.13, $8 \times 2 \times 1$ boyutlu YSA donanımı için genel bir gösterimdir. Birçok yönden farklılıklara sahip olan bu donanımların ayrıntıları ileride verilecektir.

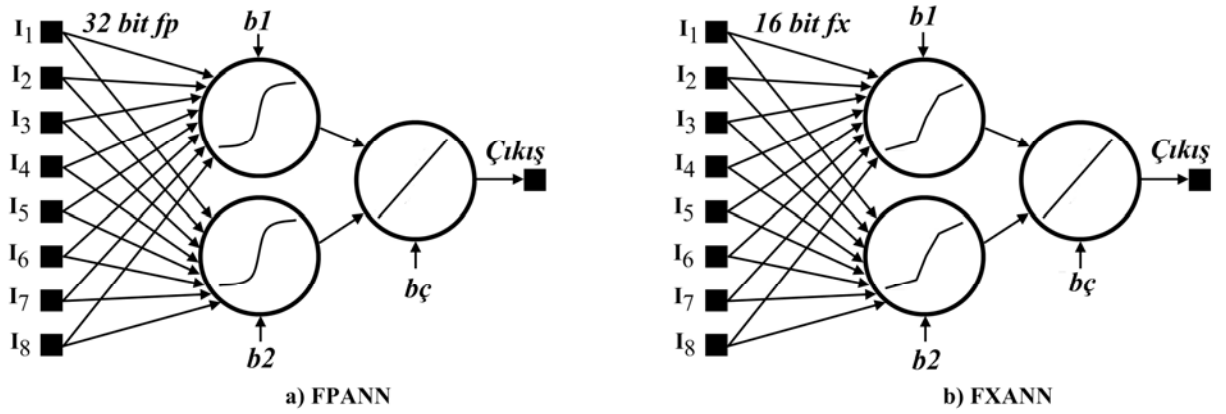


Şekil 6.13. EKG kayıtlarının FPGA modeline uygulanması.



Şekil 6.14. Sekiz girişli bir işlemci elemanın blok diyagramı.

FPANN ve FXANN için aktivasyon fonksiyonu ve veri yolu büyüklüğünü ifade eden bir gösterim Şekil 6.15'te verilmiştir. FPANN'de bütün ağ boyunca 32 bit kayan noktalı nümerik format ile işlem yapılır. FXANN ise girişlerine uygulanan 16 bit sabit noktalı nümerik formatı kullanır. Fakat çarpım terimlerinden sonra 32 bit sabit noktalı nümerik değerler oluşur. Bu değerler çarpıcıların çıkışlarında bulunan bir birim tarafından tekrar 16 bit sabit noktalı nümerik formata çevrilir. FXANN içerisindeki 32 bitlik değerlerin 16 bite dönüştürülmesinin sebebi, 16 bitlik donanımların 32 bitlik donanımlara göre daha az kaynak kullanmasıdır.



Şekil 6.15. a) FPANN b) FXANN modelleri.

8x2x1 boyutlu YSA modelleri paralel bir lojik mimari ile FPGA içerisinde tasarlanırken Altera Quartus II yazılımı kullanılmıştır. Tasarım VHDL kodları, şematik dosyalar ve MegaWizard kütüphanesinden alınan bazı IP çekirdekleri kullanılarak karma bir model halinde tasarlanmıştır. Quartus II ile tasarım yapılırken, tasarımda kullanılan dosyalardan birinin (bu dosya VHDL veya şematik olabilir) “top level entity” yani ana dosya olarak tanımlanması gerekmektedir. Bu dosya, alt birimleri ve bu birimlerin birbirleri ile olan bağlantılarını tanımlar. Bu tanımlamalar giriş çıkış terminallerinin bit genişlikleri ve içeride kullanılan sinyallerin isim ve bit genişlikleridir. Her bir alt birim yine VHDL veya şematik dosyası olabilir ve bu dosyalar özel birer isim ile kaydedilir. Tasarım akışı FPANN ve FXANN için aynı olacaktır, devam eden alt bölümde FPANN için yapılan tasarım detaylandırılmıştır.

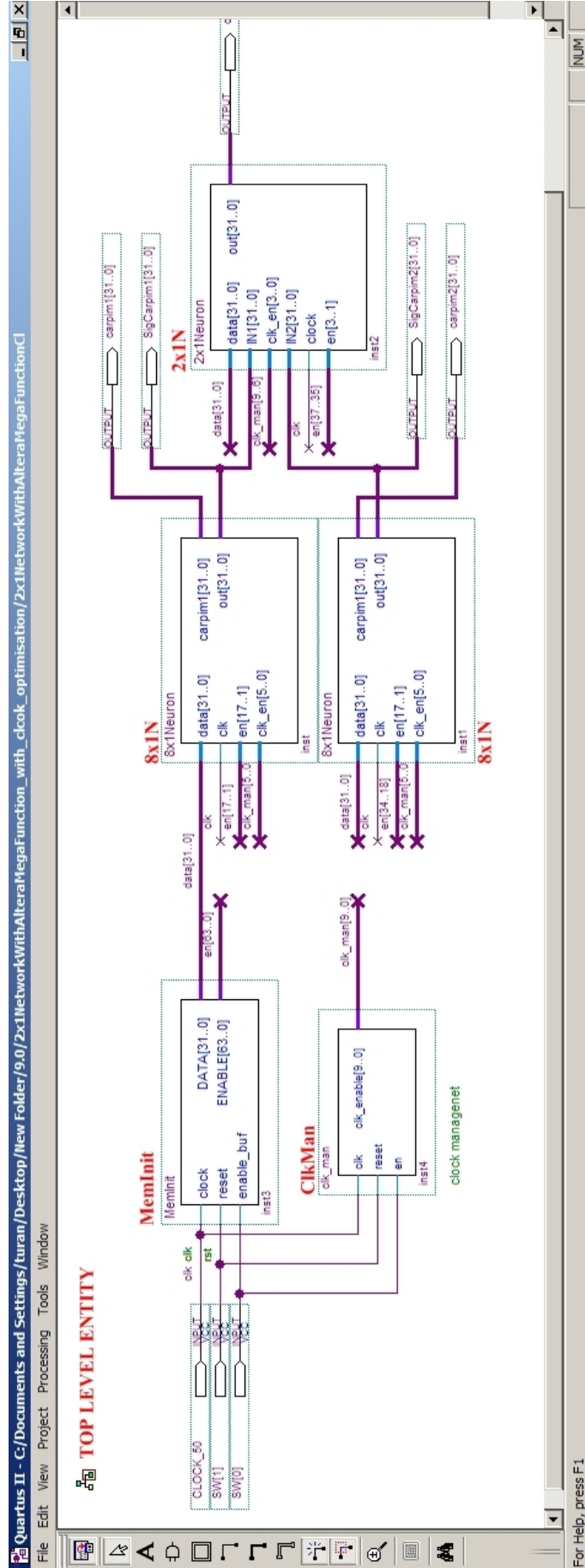
6.4.1. IEEE 754 32 Bit Kayan Noktalı Nümerik Tanımlama ile FPGA Üzerinde YSA Donanımı Gerçekleştirilmesi (FPANN)

FPANN Denklem 6.1 ile verilen doğrusal olmayan sigmoid aktivasyon fonksiyonunu kullanmaktadır. Burada net_j gizli katmandaki işlemci elemanın sinaptik toplamını, O_j ise işlemci elemanın çıkışını vermektedir. Bir başka ifade ile O_j aktivasyon fonksiyonu bloğunun çıkışıdır.

$$O_j = \frac{1}{1 + e^{-net_j}} \quad (6.1)$$

FPANN, -2^{-126} ile $+(2-2^{-23}) \times 2^{127}$ arasında değişen bir nümerik tanımlama aralığına sahip 32 bit kayan noktalı nümerik tanımlamayı kullanmaktadır. Bu nümerik formata “single precision floating point number” sistem de denilmektedir. FPAAN donanım gerçeklemesinde en zor kısım sigmoid alt bloğunun oluşturulmasıdır. FPGA üzerinde donanım modellenirken, uygun FPGA yongasının seçiminde en büyük ölçüt tasarımın ihtiyaç duyduğu donanım kaynaklarıdır. 32 bit kayan noktalı nümerik format aritmetik işlemlerin gerçekleştirilmesi için Altera FPGA yongalarındaki gömülü ve yazılımsal çarpıcıların çok sayıda kullanılması gerekecektir. Altera bu çarpıcıları DSP eleman olarak isimlendirmektedir [95, 119, 161, 164].

Altera Quartus II programı kullanılarak tasarlanmış olan FPANN’nin ana tasarım dosyası (top level entity) Şekil 6.16’da verilmiştir. Burada her bir ana birim kolaylıkla görülebilmektedir. Ayrıca FPANN tasarımının dosya organizasyonu, her bir birimin kaynak kullanımı ve bütün tasarım süreçlerinin onaylanmış olduğuna dair bilgilerin bulunduğu tasarım ekranı görüntüsü Şekil 6.17’de verilmiştir. Bu ekran görüntüsünde tasarımda kullanılan toplam lojik eleman sayısı, toplam hafıza elemanı sayısı ve toplam çarpıcı elemanların sayısı bütün tasarım ve her bir blok için ayrı ayrı verilmiştir.



Şekil 6.16. FPANN ana tasarım dosyası.

FPANN, 94'ü gizli katman işlemci elemanları içerisindeki iki adet sigmoid aktivasyon fonksiyonu bloğu ve 126'sı sistemde yer alan 18 çarpım elemanı tarafından kullanılan toplam 220 adet 9x9 (9 bitlik çarpıcı) gömülü çarpıcı birimine ihtiyaç duyar. 9x9 gömülü çarpıcıların her biri DSP eleman olarak isimlendirilir. 220 adet DSP elemanın 100 tanesi 18x18 (18 bitlik çarpıcı), 20 tanesi ise 9x9 bit boyutlu çarpıcılardan meydana gelmektedir. Her bir 8x1N birimi 103 adet DSP eleman kullanır ancak bu kaynağın 47 adet DSP kullanımı tek başına sigmoid aktivasyon fonksiyonu bloğu tarafından yapılmaktadır. Bu haliyle sigmoid aktivasyon fonksiyonu bloğu tasarımın en pahalı birimidir. İkinci en çok DSP işlemci elemanı kullanan birim ise çarpma işlemi birimidir. Her bir 32 bit kayan noktalı çarpma işlemi için 7 adet DSP işlemci eleman kullanılır. Toplayıcılar DSP işlemci elemanlarına ihtiyaç duymazlar. Fakat 32 bit kayan noktalı nümerik değerlerin toplamı için yine de çok sayıda lojik elemana ihtiyaç duyulmaktadır. FPANN ve içerisindeki dört ana bloğun donanım kaynak kullanımları toplam lojik eleman, toplam DSP işlemci elemanı ve hafıza kaynakları cinsinden, adet ve toplam kullanımın yüzdeleri cinsinden Tablo 6.4'te ayrıntılı olarak verilmiştir.

Entity	Logic Cells	Dedicated Logic Registers	DSP Elements	DSP 9x9	DSP 18x18	Pins
Cyclone III: EP3C120F780C8						
2x1NetworkWit...	23189 (0)	10816 (0)	220	20	100	163
8x1Neuron.inst	10369 (0)	4832 (0)	103	9	47	0
8x1Neuron.inst1	10320 (0)	4814 (0)	103	9	47	0
2x1Neuron.inst2	2119 (0)	1054 (0)	14	2	6	0
MemInit.inst3	125 (0)	75 (0)	0	0	0	0
clk_man.inst4	256 (256)	41 (41)	0	0	0	0

Task	Time
Compile Design	00:18:44
Analysis & Synthesis	00:06:59
Fitter (Place & Route)	00:06:54
Assembler (Generate programming files)	00:00:35
TimeQuest Timing Analysis	00:02:30
EDA Netlist Writer	00:01:46
Program Device (Open Programmer)	

Şekil 6.17. Kaynak kullanımları ve tasarım süreçlerinin onayı (FPANN).

Tablo 6.4. FPANN ve ana bloklarının donanım kaynak kullanımları.

	2* 8x1N	2x1N	MemInit	ClockMan	FPANN
Toplam Lojik Eleman	20.689	2.119	125	256	23.189
	89%	9%	$\leq 1\%$	$\leq 1\%$	100%
Toplam DSP Eleman	206	14	0	0	220
	94%	6%			100%
Toplam Hafıza Elemanı	9.646	1.054	75	41	10.816
	89%	10%	$\leq 1\%$	$\leq 1\%$	100%

**FPANN iki adet 8x1 bloğuna sahiptir*

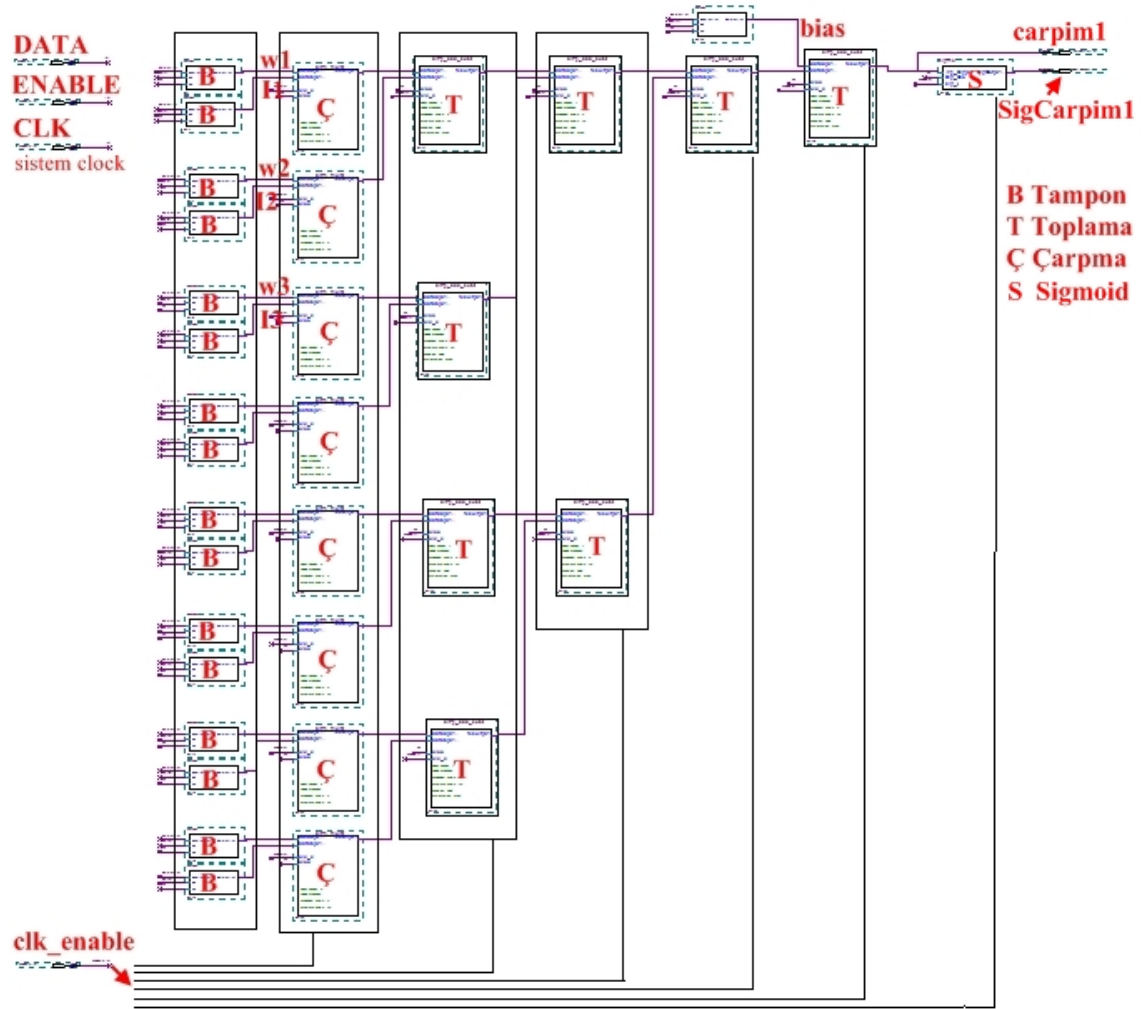
Her bir 8x1N ana bloğu birer adet doğrusal olmayan sigmoid aktivasyon fonksiyonu alt bloğu ve her bir girişin ağırlıklarla çarpılmasını sağlayan sekizer adet çarpıcıya sahip olduğu için FPANN tasarımının en çok kaynak kullanan birimidir. FPANN 32 bit kayan noktalı nümerik formatla işlem gördüğü için çarpıcılar ve toplayıcılar 32x32 boyutludur. Her bir aritmetik işlem bloğunun çıkışı da 32 bit kayan noktalı nümerik formattadır. Çıkış işlemci elemanında yer alan purelin aktivasyon fonksiyonu $y = x$ şeklinde doğrusal bir cevaba sahip olduğundan direk bağlantıyı ifade eder. Bu yüzden herhangi bir donanım kaynağı kullanmaz. Tablo 6.5'te sigmoid, purelin aktivasyon fonksiyonları ve çarpıcı, toplayıcı birimlerinin kaynak kullanımları, toplam lojik eleman, toplam DSP eleman ve toplam hafıza elemanı cinsinden detaylı olarak verilmiştir.

Tablo 6.5. FPANN içerisindeki bazı önemli birimlerin kaynak kullanımları.

	Sigmoid	Çarpıcı	Purelin	Toplayıcı
Toplam Lojik Eleman	1.942	259	0	716
Toplam DSP Eleman	47	7	0	0
Toplam Hafıza Elemanı	892	203	0	283

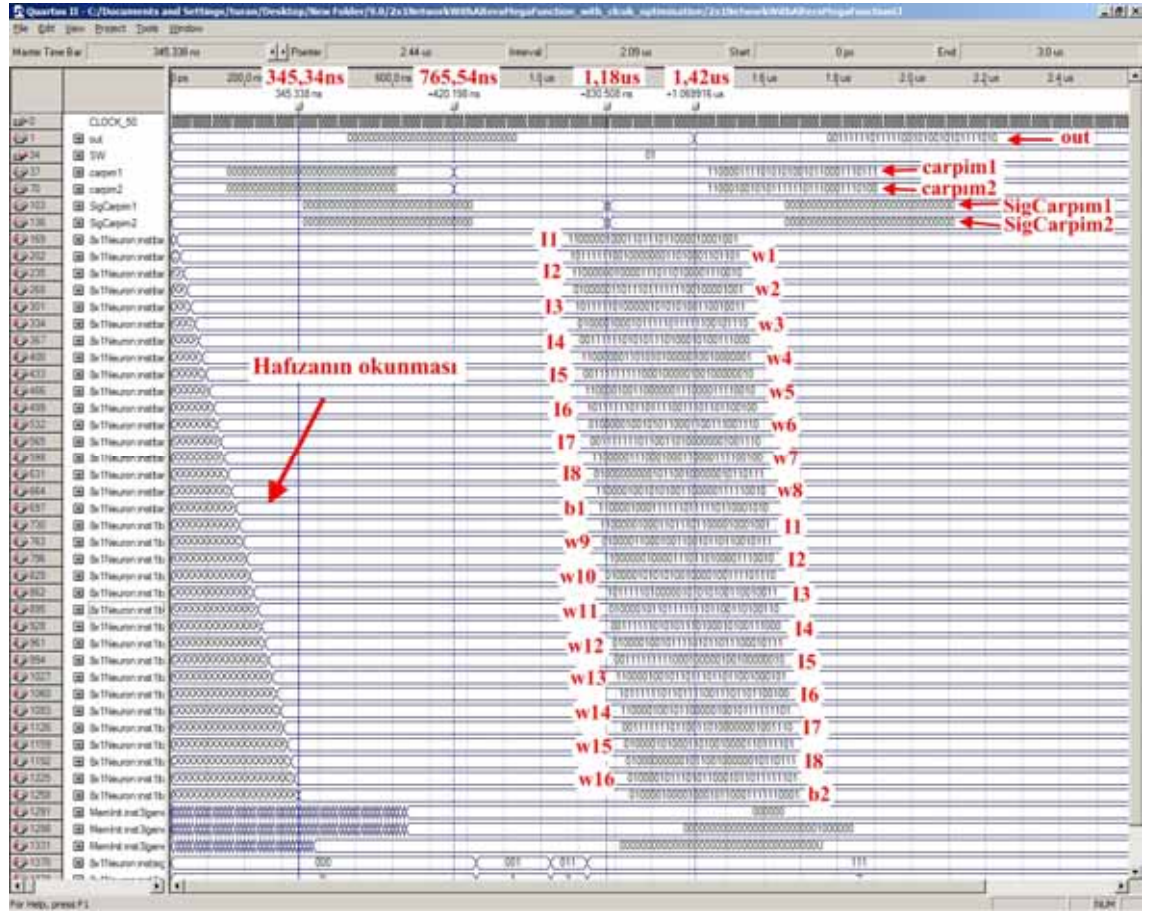
Tasarımın ana bloklarına kısaca göz atılacak olursa, Şekil 6.18'de görüldüğü gibi MemInit bloğu içerisinde 32 bitlik 64 adresli bir hafıza (lpm_rom0 alt bloğu) bulunur ve bu hafıza içerisinde bulunan ağırlık ve giriş değerleri generator alt bloğu tarafından üretilen adres bilgisi ile DATA hattına verilir. DATA hattı bütün ağ içerisine dağılır ve bu hat üzerindeki bilgi 8x1N ve 2x1N blokları içerisindeki tampon alt birimleri tarafından okunur. Hangi tampon'un DATA hattını ne zaman okuyacağı ENABLE çıkışı ile kontrol edilmektedir. Bu çıkış, birim zamanda sadece bir tampon'u aktif etmektedir.

devreye alınması ve çıkarılmasını sağlar. Her bir blok tanımlı bir clock periyodu boyunca işlem görmektedir ve bu işlem süresi daha önceden hesaplanarak Şekil 6.16'da verilen clk_man bloğuna VHDL olarak kodlanmıştır. Bu sayede ilgili blok sistemin clock kontrolünü sağlar. Bu işaret, sistemdeki verinin paralel olarak işlenmesini sağlamaktadır. DATA ve CLK bütün tampon bloklarına uygulanır ancak, Şekil 6.19'da görüldüğü gibi ENABLE sinyali birim zamanda ancak bir tamponun DATA bilgisini hafızaya almasına müsaade eder. Böylece hafızadaki ağ parametreleri ve giriş değerleri doğru bir şekilde çekilerek, ağırlıklar ilgili giriş değerleri ile çarpılmak üzere çarpma birimlerine uygulanır.



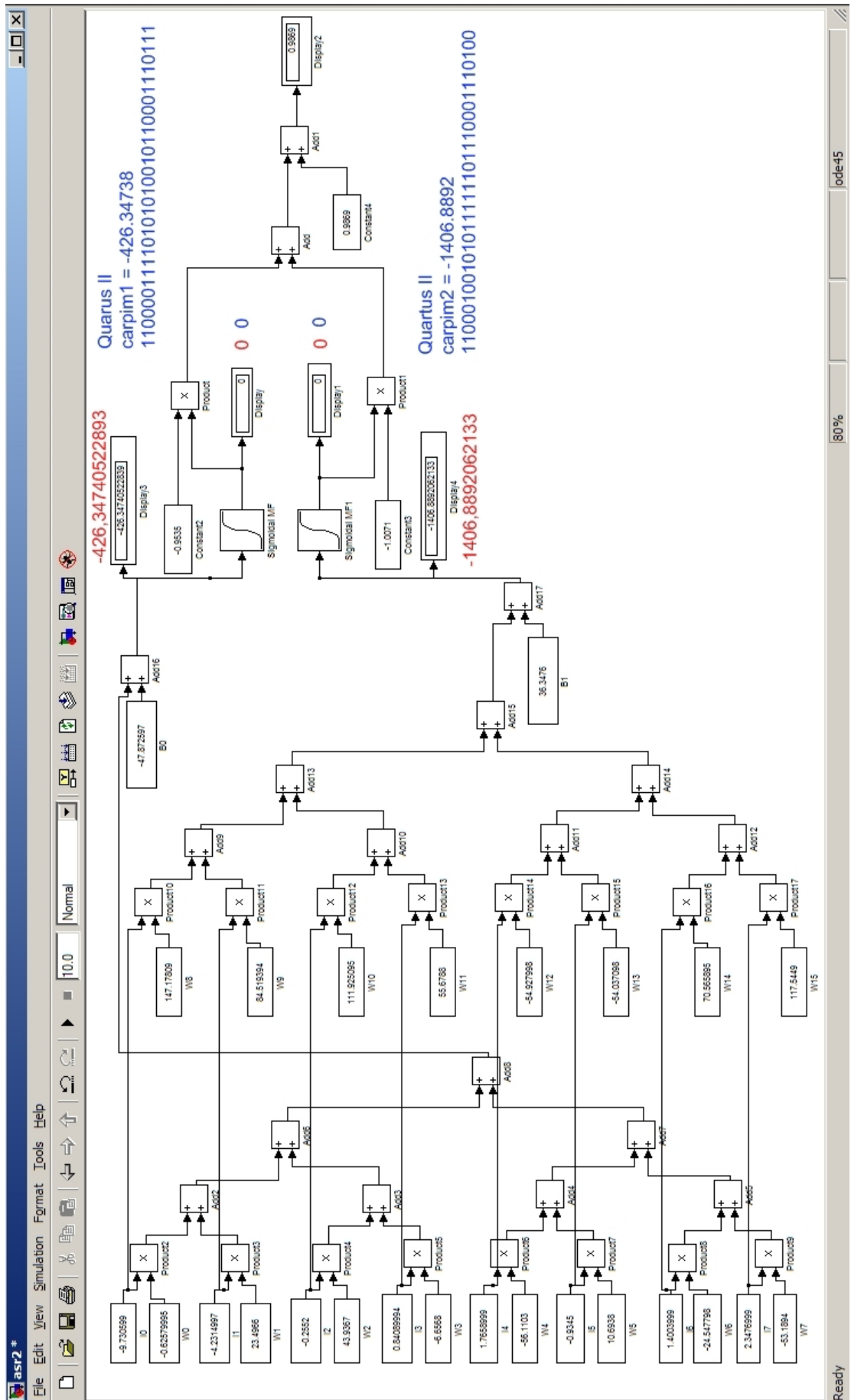
Şekil 6.20. 8x1N bloğunun içyapısı.

Şekil 6.21'de 8x1N bloklarının çıkışlarının aynı anda üretildiği ve bu çıkışların sigmoid aktivasyon fonksiyonu bloğu tarafından yorumlanması için de belli bir süre geçmesi gerektiği görülmektedir. Bütün bu işlem sırası ve süresi clk_man bloğu tarafından üretilen clk_enable tarafından organize edilmektedir.



Şekil 6.21. FPANN donanımının fonksiyonel benzetimi.

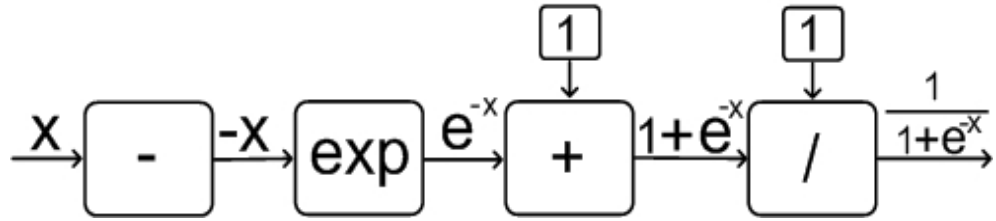
Şekil 6.21’de görüldüğü gibi veriler hafızadan düzenli bir şekilde çekilerek okunmuş ve bu işlem 346 ns sürmüştür. Hafızadan okunan verilerin yani ağırlık ve giriş değerlerinin çarpılarak sonuçlarının toplanması ve bu ağırlıklandırılmış giriş değerlerinin (carpim1 ve carpim2) sigmoid aktivasyon fonksiyonu bloğunun girişine uygulanması işlemi 420 ns sürmüştür. Sigmoid aktivasyon fonksiyonu bloğunun, ağırlıklandırılmış girişleri yorumlaması 410 ns sürmüştür. Gizli katman nöronlarının (8x1N) çıkışları (carpim1 ve carpim2) çıkış nöronuna (2x1N) uygulanmış ve bu katmanın ağırlık çıkışını üretmesi 240 ns sürmüştür. FPANN ağ donanımı 50 MHz clock frekansı ile 1.42 μ s’de hafızadan ağ parametreleri ve 8 giriş verisini okuyarak cevap üretmiştir. FPANN giriş sayısı sabit tutularak gizli katmandaki nöron sayısı arttırılırsa donanımın cevap süresi değişmeyecektir. Görüldüğü gibi carpim1 ve carpim2 değerleri aynı anda elde edilmiştir, çünkü paralel mimari artan nöron sayısından etkilenmemektedir. Bu benzetim sonuçları Şekil 6.22’de verilen aynı ağırlık ve giriş değerlerine sahip bir Simulink modeli ile karşılaştırılmış ve sonuçların aynı olduğu görülmüştür.



Şekil 6.22. F-ANN Simulink modeli.

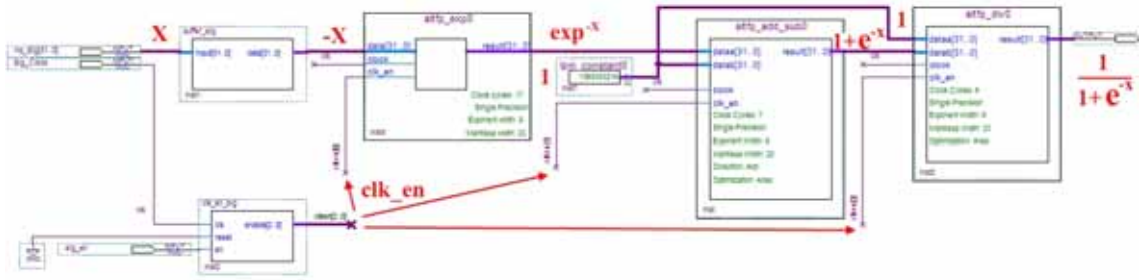
Şekil 6.22’de carpim1 ve carpim2 sinyalleri için Simulink modelinin ürettiği sonuçlar kırmızı, Quartus II benzetim editörünün ürettiği sonuçlar ise mavi ile yazılmıştır. Simulink ve Quartus II sonuçlarının arasında çok küçük bir fark var. Bu fark yuvarlama işleminden kaynaklanmaktadır.

Sigmoid aktivasyon fonksiyonu bloğunun içyapısı Şekil 6.23’te diyagram olarak verilmiştir. Burada $8 \times 1N$ giriş ve ağırlık değerlerinin çarpımlarının toplamı (örn carpim1) negatife çevrilerek (x giriş değerinden $-x$ değeri elde edilir) exponansiyel işlem bloğuna uygulanır ve $y = \exp(-x)$ sonucu üretilir. Bu değer daha sonra sabit 1 ile toplanarak $1 + \exp(-x)$ değeri üretilir ve bu değer sabit 1’e bölüm olarak uygulanır. Sigmoid aktivasyon fonksiyonu bloğu üst alma işleminden dolayı çok fazla kaynak kullanmaktadır.



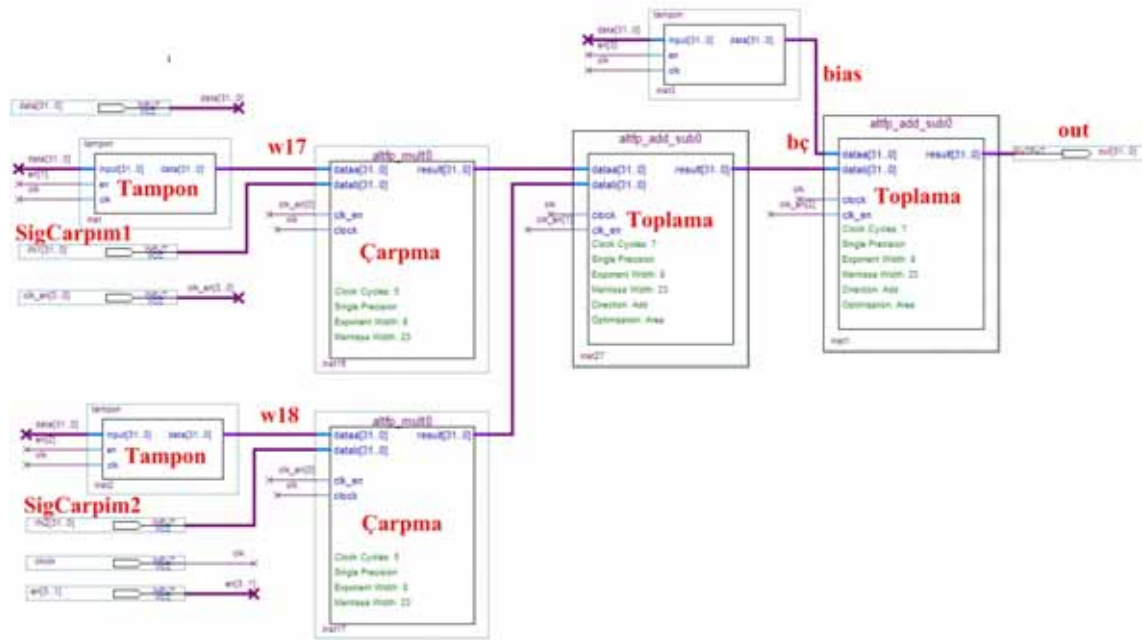
Şekil 6.23. Sigmoid aktivasyon fonksiyonu bloğu blok diyagramı.

$8 \times 1N$ bloğu içerisinde yer alan sigmoid aktivasyon fonksiyonu bloğunun şematik tasarım dosyası Şekil 6.24’te verilmiştir. Burada X girişinin uygulandığı tampon bloğu, giriş ifadesinin işaret bitini değiştirerek 32 bit kayan noktalı nümerik ifadenin – (eksi) ile çarpımını sağlar. Üst alma operatörü Altera Mega Function kütüphanesinden alınarak kullanılmıştır.



Şekil 6.24. Sigmoid aktivasyon fonksiyonu bloğunun içyapısı.

Daha önce de bahsedildiği gibi sistem içerisindeki 8x1N bloklarının çıkışlarının uygulandığı 2x1N bloğunda aktivasyon fonksiyonu olarak purelin $y = x$ fonksiyonu kullanılmıştır. Bu fonksiyon girişi hiçbir değişiklik olmadan çıkışa aktarmaktadır yani sadece bir bağlantıdır. Şekil 6.25'te 2x1N bloğunun içyapısı şematik olarak verilmiştir. Burada tamponlarda tutulan ağırlık değerleri 8x1N bloğu tarafından üretilen çıkış değerleri ile çarpılır, bu verilerin toplamı bir bias değeri ile toplanarak çıkışa aktarılır. 2x1N bloğu içerisindeki alt birimlerin çalışması yine clk_enable sinyali tarafından kontrol edilmektedir.



Şekil 6.25. 2x1N bloğunun içyapısı.

32 bit kayan noktalı nümerik tanımlama formatı aritmetik işlemlerde çok yüksek derecede doğrulukla cevap üretirken, donanım gerçeklemelerinde kaynak kullanımını önemli ölçüde arttırmaktadır. Kaynak kullanımını arttırmasına rağmen yüksek hassasiyetin gerektiği uygulamalarda kayan noktalı nümerik tanımlama formatı sıklıkla tercih edilmektedir. FPANN EP3C120F780 yongasının toplam lojik elemanlarının %19'unu ve gömülü çarpıcılarının ise %38'ini kullanmıştır. Bir sonraki başlıkta FXANN tasarımı incelenecek ve daha sonra bu iki yapı kaynak kullanımları bakımından incelenecektir.

6.4.2. 16 Bit Sabit Noktalı Nümerik Tanımlama ile FPGA Üzerinde YSA Donanımı Gerçekleştirilmesi (FXANN)

FXANN işaret bitli 16 bit sabit noktalı nümerik tanımlamayı kullanmaktadır. 16 bitin ilk biti işaret biti, takip eden yedi biti tamsayı değeri ve kalan sekiz biti ise virgülden sonraki hassasiyeti tanımlamak için kullanılmıştır. 16 bit nümerik tanımlama ± 128 nümerik aralığını 2^{-8} hassasiyet ile tanımlamaktadır. Bu nümerik tanımlama normalize edilmiş veri setinin hesaplamasını sağlayacak yeterlilikte bir nümerik aralığa sahiptir. FXANN donanımının Matlab modelinde elde edilen ağ parametreleri Tablo 6.6'da verilmiştir. 5 adet kalp atımı için TBA işlemi sonucunda elde edilen ilk sekiz temel bileşenin (TB) nümerik değerleri ise Tablo 6.7'de verilmiştir. Temel bileşenler ve ağ parametrelerinin çarpım veya toplamlarından hesaplanacak nümerik değerlerin 16 bit sahasında ifade edileceği ve nümerik değerlerde taşmalar oluşmayacağı açıkça görülmektedir.

Tablo 6.6. 8x2x1 boyutlu YSA için ağ parametreleri FPANN.

$w1$ = -1,6323	$w9$ = -3,8596	$w16$ = -2,1545
$w2$ = -3,0263	$w10$ = -1,5485	$w17$ = 1,7458
$w3$ = -1,3357	$w11$ = 1,2032	$b1$ = 0,8821
$w4$ = 1,1289	$w12$ = -1,5076	$b2$ = -8,4431
$w5$ = 3,8375	$w13$ = 0,9262	$bç$ = 0,9886
$w6$ = -3,2129	$w14$ = 2,5759	
$w7$ = 0,9284	$w15$ = -0,3038	
$w8$ = 0,5648	$w16$ = -2,0272	

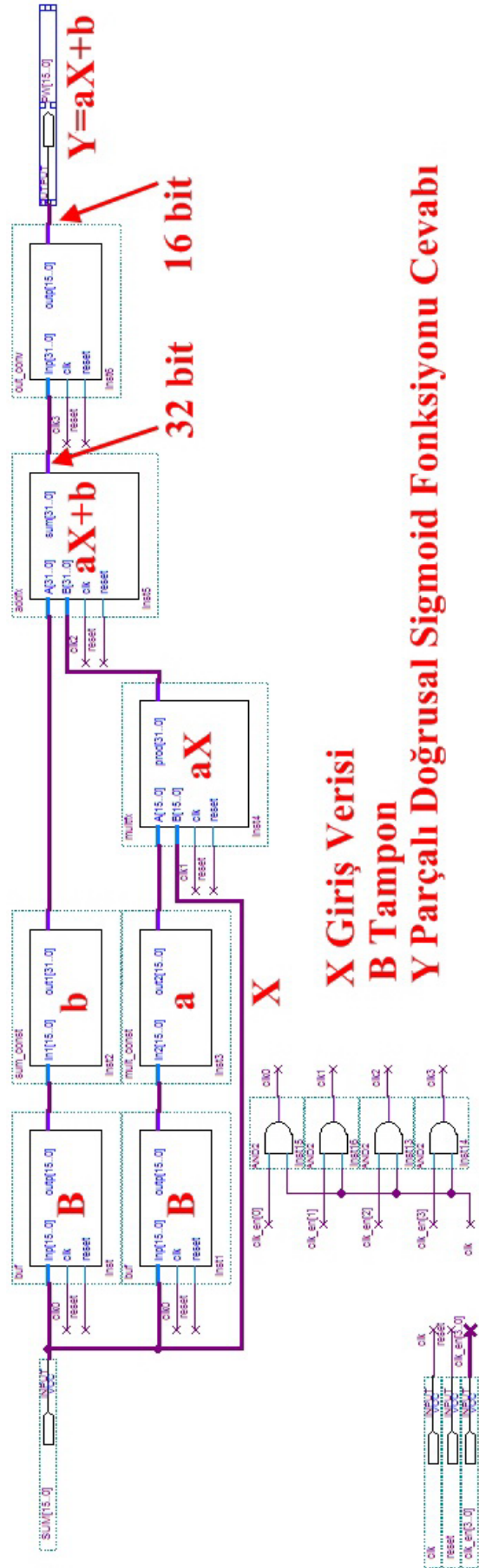
Tablo 6.7. İlk sekiz temel bileşen.

TB'ler	1. Kalp Atımı	2. Kalp Atımı	3. Kalp Atımı	4. Kalp Atımı	5. Kalp Atımı
TB1	-0,3249	-0,9469	-0,2755	-0,4913	-1,1727
TB2	-0,3337	0,5849	-0,5448	-0,0604	1,0796
TB3	-0,0275	-0,0428	-0,0092	-0,0789	-0,0963
TB4	-0,0731	-0,1485	0,0375	-0,0978	-0,1431
TB5	-0,0794	-0,01397	-0,0620	-0,0138	0,0574
TB6	0,1506	0,0821	0,1778	0,1621	0,0862
TB7	-0,0815	-0,0582	-0,0122	-0,1038	-0,2057
TB8	-0,3379	-0,3044	-0,3471	-0,3450	-0,2335

FXANN donanımı aktivasyon fonksiyonu olarak Parçalı Doğrusal Sigmoid Aktivasyon (PWLAN) fonksiyonunu kullanmaktadır. 16 bit sabit noktalı nümerik tanımlama doğrusal olmayan aritmetik işlemlerin hesaplanmasında direk olarak kullanılamamaktadır. Bu yüzden sigmoid aktivasyon fonksiyonu bir eğri uydurma yöntemi olan PWLAN ile yedi parçalı doğrusal olarak Denklem 6.2’de verildiği gibi formüle edilmiştir. Burada $f(X)$ PWLAN aktivasyon fonksiyonunun çıkış cevabıdır ve 0 ile 1 arasında değerler alır. FPANN için PWLAN kuralı, VHDL dil editörü kullanılarak, 16 bit sabit noktalı nümerik format ile FPGA üzerinde oluşturulmuştur.

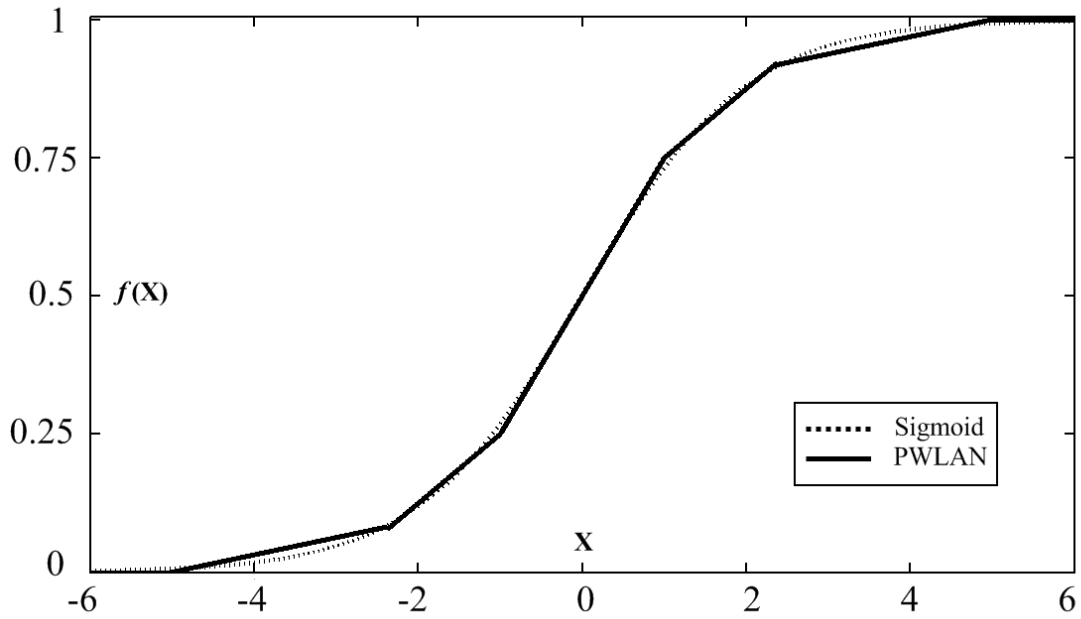
$$\begin{array}{ll}
 f(X) = 1 & 5 \leq X \\
 f(X) = 0,03125X + 0,84375 & 2,375 \leq X < 5 \\
 f(X) = 0,125X + 0,625 & 1 \leq X < 2,375 \\
 f(X) = 0,25X + 0,5 & -1 < X < 1 \\
 f(X) = 0,125X + 0,375 & -2,375 < X \leq -1 \\
 f(X) = 0,03125X + 0,15625 & -5 < X \leq -2,375 \\
 f(X) = 0 & X \leq -5
 \end{array} \quad (6.2)$$

8x2x1 boyutlu YSA modeli, FXANN gerçeklemesinden önce yukarıda denklemini verilen PWLAN aktivasyon fonksiyonu kullanılarak Matlab ortamında eğitilmiştir. Çünkü parçalı doğrusal yaklaşım tam olarak sigmoid fonksiyonunun cevabını vermediği için yapay sinir ağının eğitimin PWLAN ile yapılması gereklidir. Aksi takdirde eğitim hatasına ilave olarak bu iki fonksiyonun fark hatası da YSA’nın sınıflandırma hatasına eklenecektir. PWLAN fonksiyonu Matlab ortamında bir komut dosyası olarak oluşturulduktan sonra eğitim esnasında çağrılarak 8x2x1 boyutlu referans YSA’nın eğitiminde kullanılmıştır. $f(X) = aX+b$ şeklinde yedi parça olarak ifade edilen PWLAN fonksiyonu Quartus II’de Şekil 6.26’daki gibi oluşturulmuştur.



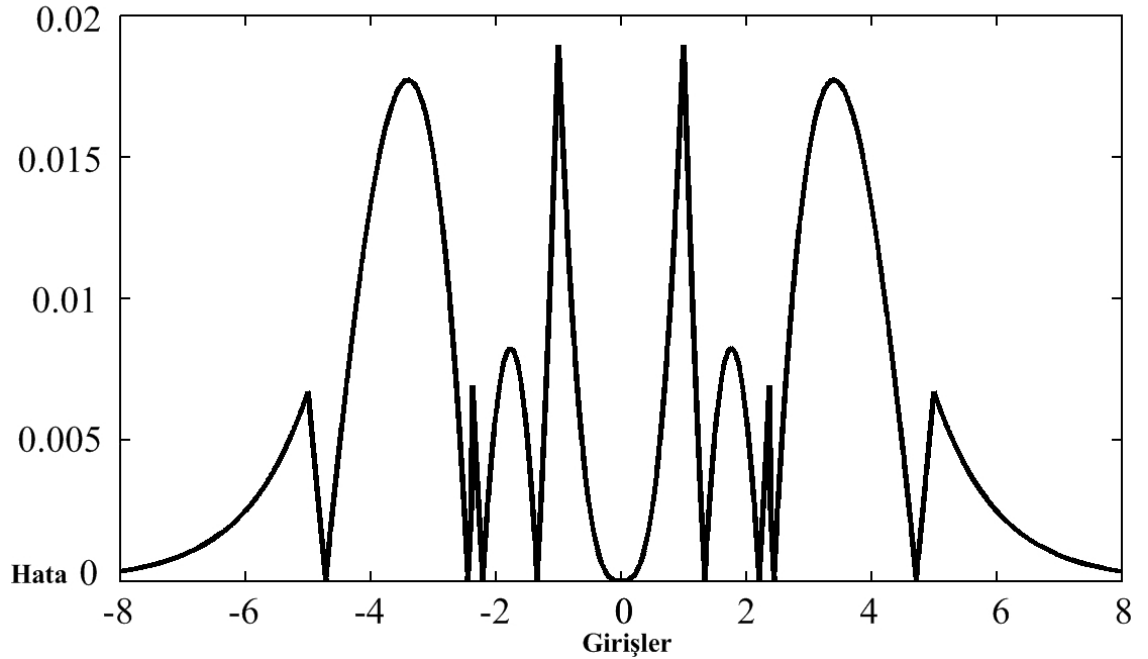
Şekil 6.26.WLAN bloğunun içyapısı.

PWLAN yedi parçalı yaklaşım ile $[-8 \ 8]$ aralığı için X giriş verisine bağlı olarak PWLAN çıkış cevabı maksimum %1,89 hata ile gerçek sigmoid fonksiyonunun çıkış cevabına yakınsamaktadır. Bu değer %2'nin altında kabul edilebilir bir hata oranıdır. Gerçek sigmoid fonksiyonu ile parçalı doğrusal yaklaşım arasındaki fark hatası, bu çalışmada yedi adet kullanılan aralıkların sayısının artırılması ile daha da azaltılabilir. PWLAN fonksiyonunun $[-8 \ 8]$ aralığında çıkış karakteristiği ve gerçek sigmoid fonksiyonu ile arasındaki fark hatasının grafiksel gösterimleri sırası ile Şekil 6.27 ve Şekil 6.28'de verilmiştir.



Şekil 6.27. PWLAN fonksiyonunun gerçek sigmoide yakınsaması.

FPANN donanımında kullanılan gerçek sigmoid ile FXANN donanımında kullanılan PWLAN aktivasyon fonksiyonlarının birbirlerinden çıkarılması ile elde edilen fark Şekil 6.28'de görüldüğü gibi %2'den azdır. X giriş verisi eksen $[-8 \ 8]$ aralığında verilmiştir. Çünkü bu aralığın dışındaki giriş değerleri logaritmik sigmoid fonksiyonunu sırası ile 0 ve 1 değerlerine satüre etmektedir. Bu yüzden tanımlı aralığın dışındaki değerler için hata oranı %0 civarlarındadır. Eğitimler esnasında kullanılan parçalı doğrusal sigmoid aktivasyon fonksiyonunun Matlab kodları Ekler kısmında verilmiştir.



Şekil 6.28. PWLAN fonksiyonunu ile gerçek sigmoid arasındaki fark.

PWLAN ile gerçek sigmoid aktivasyon fonksiyonu arasındaki karakteristik fark oldukça küçük olmasına karşın, PWLAN aktivasyon fonksiyonu oldukça az yonga alanı kullanmaktadır. Kaynak kullanımının azalmasında 16 bit nümerik tanımlamanın çarpma ve diğer aritmetik işlemleri basitleştirmesinin etkisinin yanında, PWLAN fonksiyonunda bölme ve üst alma işlemlerinin olmamasının da bu azalmada büyük bir katkısı vardır. Tablo 6.8’de FXANN donanımında kullanılan PWLAN ve purelin aktivasyon fonksiyonları ile çarpıcı ve toplayıcı elemanların kaynak kullanımları verilmiştir. Tablodan görüleceği üzere PWLAN aktivasyon fonksiyonu birimi sadece iki adet DSP elemanı kullanmaktadır. Bu iki adet DSP elemanı ise PWLAN birimi içerisinde yer alan çarpma birimi tarafından kullanılmaktadır.

Tablo 6.8. FXANN içerisindeki bazı önemli birimlerin kaynak kullanımları.

	PWLAN	Çarpıcı	Purelin	Toplayıcı
Toplam Lojik Eleman	73	0	16	32
Toplam DSP Eleman	2	2	0	0
Toplam Hafıza Elemanı	42	0	16	32

FXANN donanımının kaynak kullanımı ve onaylanmış tasarım akışının bulunduğu ekran görüntüsü Şekil 6.29’da verilmiştir. FPANN ve FXANN aynı mimarinin iki farklı nümerik tanımlama ile oluşturulmuş halleri olduğundan ağı oluşturan bloklar aynıdır.

The screenshot shows the Quartus II interface with the Project Navigator and Tasks windows. The Project Navigator displays the resource usage for the Cyclone III EP3C120F780C7 device. The Tasks window shows the compilation process, including Compile Design, Analysis & Synthesis, Fitter (Place & Route), Assembler (Generate programming files), TimeQuest Timing Analysis, and EDA Netlist Writer.

Entity	Logic Cells	Dedicated Logic Registers	DSP Elements	DSP 9x9	DSP 18x18	Pins
Cyclone III: EP3C120F780C7						
8x2x1_16Bitfxp	1814 (0)	784 (0)	40	0	20	124
Mem_Init.inst	129 (0)	75 (0)	0	0	0	0
clk_man.inst1	269 (269)	52 (52)	0	0	0	0
8x1Neuron.inst2	638 (6)	298 (0)	18	0	9	0
8x1Neuron.inst3	639 (6)	298 (0)	18	0	9	0
2x1Neuron.inst4	142 (4)	61 (0)	4	0	2	0

Task	Time
Compile Design	00:02:43
Analysis & Synthesis	00:00:46
Fitter (Place & Route)	00:01:04
Assembler (Generate programming files)	00:00:21
TimeQuest Timing Analysis	00:00:21
EDA Netlist Writer	00:00:11
Program Device (Open Programmer)	

Şekil 6.29. Kaynak kullanımları ve tasarım süreçlerinin onayı (FXANN).

FXANN 4'ü gizli katman işlemci elemanlarında bulunan aktivasyon fonksiyon blokları ve 36'sı sistemde yer alan 18 çarpım elemanı tarafından kullanılan toplam 40 DSP işlem elemanına ihtiyaç duymaktadır. Sonuç olarak 40 DSP işlemci elemanının 36 tanesi 8x1N bloklarında 4 tanesi ise 2x1N bloğunda kullanılır. FXANN tasarlanırken gizli katmanlarda kullanılan aktivasyon fonksiyonları, yedi parçalı doğrusal PWLAN sigmoid fonksiyonu olarak seçildiği daha önce de ifade edilmişti. Her bir PWLAN iki adet DSP işlemci elemanı kullanır. Bu iki DSP işlem elemanı, PWLAN bloğu içerisinde bulunun bir çarpıcı tarafından kullanılmaktadır. 8x2x1 boyutlu YSA modelinin 16 bit nümerik tanımlama kullanılarak FPGA üzerinde tasarlanması ile kaynak kullanımı çok önemli ölçüde azaltılmıştır. Tablo 6.9'da, FXANN ve içerisindeki dört ana bloğun donanım kaynak kullanımları, toplam lojik eleman, toplam DSP işlemci elemanı ve hafıza kaynakları cinsinden, adet ve toplam kullanımın yüzdeleri cinsinden ayrıntılı olarak verilmiştir.

Tablo 6.9. FXANN ve ana bloklarının donanım kaynak kullanımları.

	2 8x1N	2x1N	MemInit	ClockMan	FXANN
Toplam Lojik Eleman	1.277 70%	142 8%	129 7%	269 15%	1.814 100%
Toplam DSP Eleman	36 90%	4 10%	0	0	40 100%
Toplam Hafıza Elemanı	596 75%	61 8%	75 10%	52 7%	784 100%

**FXANN iki adet 8x1 bloğuna sahiptir*

Tablo 6.5 ve Tablo 6.8 karşılaştırıldığında FPANN daha fazla donanım kaynağı tüketiyor olmasına karşın, purelin aktivasyon fonksiyonu için hiç lojik, DSP veya hafıza elemanı kullanmadığı halde FXANN 16'şar adet lojik ve hafıza elemanı kullanmıştır. Bunun sebebi $y=x$ karakteristiğindeki purelin fonksiyonu için herhangi bir donanıma ihtiyaç duyulmamasıdır. FPANN donanımında 32 bit nümerik değer bu bloğa geldiği haliyle çıkışına transfer edilir. Yani bu blok ileri yönlü bir bağlantıdır. FXANN donanımında kullanılan lojik ve hafıza elemanlarının sebebi ise çıkış katmanındaki işlemci elemanda yani 2x1N bloğunda, 16 bitlik ağırlık değerleri ile gizli katmandaki 8x1N bloklarından gelen 16 bitlik giriş değerlerinin çarpılması sonucunda oluşan 32 bitlik nümerik ifadenin 16 bit ile ifade edilmesi için kullanılan bir alt birimdir. Bu çevrim işlemi sonucunda eğer taşma oluşursa bir hata kodu üretilerek işlem sonlandırılır ve bir diğer atım için hesaplama yapılır.

16 bit ile tanımlamayacak kadar büyük olan değerler ± 128 değerine çevrilerek kullanıldığında da sonuçlar genellikle değişmemekte ve doğru çıkmaktadır. Çünkü sigmoid aktivasyon fonksiyonu $[-6 \text{ ile } 6]$ arasında duyarlıdır ve $[-8 \text{ ile } 8]$ aralığının dışına çıkıldığında, sigmoid fonksiyonunun çıkışı $[0 \text{ veya } 1]$ değerine satüre olmaktadır. ± 128 değerleri saturasyon bölgelerindedir ve bu değerlerin ürettikleri sonuçlar genellikle doğru çıkmaktadır.

6.5. FPANN ve FXANN Donanımlarının Kaynak Kullanımları Bakımından Karşılaştırılması

16 bit nümerik tanımlama ve parçalı doğrusal sigmoid aktivasyon fonksiyonunun kullanılması ile kaynak kullanımı başka bir ifade ile donanım maliyeti çok büyük ölçüde azaltılmıştır. Tablo 6.10'da FPANN ve FXANN donanımları ve bu donanımlar içerisinde yer alan temel bloklarının kaynak kullanımları karşılaştırmalı olarak verilmiştir. Bu tablonun en son sütununda yer alan boyut sekmesi, 16 bit nümerik tanımlama ve PWLAN aktivasyon fonksiyonunun kullanılması ile FXANN donanımının ne ölçüde küçültüldüğü hakkında çarpıcı sonuçlar vermektedir. FXANN donanımı toplam lojik eleman sayısı bakımından 12,78, toplam DSP eleman sayısı bakımından 5,5 ve toplam hafıza elemanı sayısı bakımından 13,8 defa FPANN donanımından daha küçüktür.

Tablo 6.10. Temel blokların kaynak kullanımları bakımından karşılaştırılması.

		8x1N*	2x1N	MemInit	ClockMan	Toplam	Boyut
Toplam Lojik	FPANN	20.690	2.118	125	256	23.189	1 br*
Eleman	FXANN	1.276	142	129	268	1.814	0,078 br
Toplam DSP	FPANN	206	14	0	0	220	1 br
Eleman	FXANN	36	4	0	0	40	0,18 br
Toplam Hafıza	FPANN	9.646	1.054	75	41	10.816	1 br
Elemanı	FXANN	596	61	75	52	784	0,072 br

*8x1N bloğu iki adet kullanılmıştır ve tabloda verilen değer toplamı ifade eder.

*br “birim” kelimesinin kısaltması olarak kullanılmıştır. FPANN tasarımı 1 birim alanda gerçekleştirilen donanımların FXANN tasarımı kaç birimlik alana sığdırıldığı vurgulanmak istenmiştir.

Nümerik tanımlamalarda 32 bit yerine 16 bit kullanılması, işlem gören veri boyutunu azalttığı için FXANN donanımında kullanılan toplam lojik ve hafıza elemanlarında çok ciddi ölçüde azalma meydana gelmiştir. Veri boyutundaki azalma DSP eleman sayısına da etki etmekle beraber, DSP eleman sayısındaki azalmanın en önemli sebebi sabit noktalı nümerik hesaplama algoritmasının, kayan noktalı nümerik hesaplama algoritmasına göre daha basit olmasıdır. Ayrıca PWLAN fonksiyonunun karmaşık sigmoid fonksiyonuna göre daha basit bir aritmetik formülizasyona sahip olması da donanım kaynak kullanımlarının azalması yönünde oldukça etkili olmuştur. Bütün bu etkiler FXANN donanımının daha basit olması ve az sayıda yonga alanı kullanması yönünde etkili olmuşlardır.

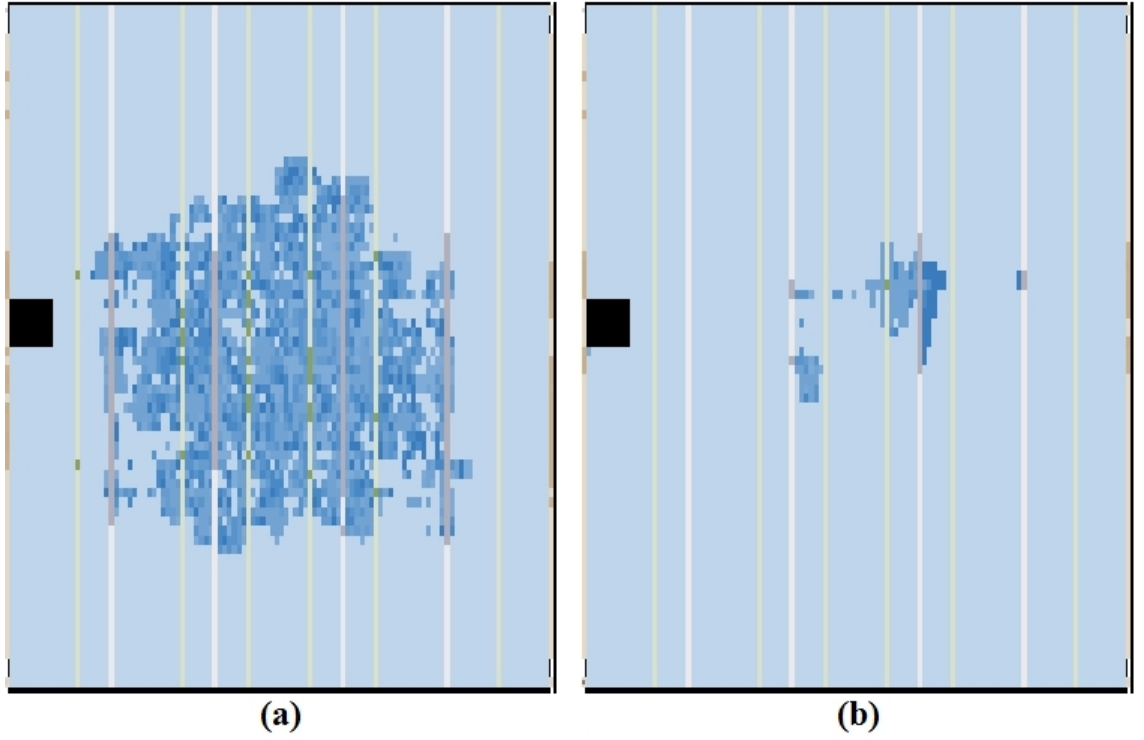
PWLAN fonksiyonu ve 16 bit nümerik sayı formatının aritmetik birimler üzerinde kaynak kullanımına olan etkisinin daha iyi anlaşılabilmesi için aritmetik birimlerin kaynak kullanımları Tablo 6.11’de detaylı bir şekilde verilmiştir. Bu tabloda kaynak kullanımına ciddi etkileri olan aktivasyon fonksiyonları (Sigmoid ve PWLAN), çarpma bloğu, sayı formatlarından çok fazla etkilenmeyen purelin aktivasyon fonksiyonu bloğu ve toplama alt bloğu kaynak kullanımları bakımından incelenmiştir. Bu sayede nümerik sayı formatlarının hangi aritmetik işlemlere daha duyarlı olduğu da görülebilir.

Tablo 6.11. Aritmetik birimlerin kaynak kullanımları bakımından karşılaştırılması.

			AF*	Çarpıcı	Purelin	Toplayıcı
Toplam Lojik	FPANN	<i>Sigmoid</i>	1.942	259	0	716
Eleman	FXANN	<i>PWLAN</i>	73	0	16	32
Toplam DSP	FPANN	<i>Sigmoid</i>	47	7	0	0
Eleman	FXANN	<i>PWLAN</i>	2	2	0	0
Toplam Hafıza	FPANN	<i>Sigmoid</i>	892	203	0	283
Elemanı	FXANN	<i>PWLAN</i>	42	0	16	32

*AF gizli katmandaki işlemci elemanlarda kullanılan aktivasyon fonksiyonudur. AF FPANN için sigmoid FXANN için PWLAN aktivasyon fonksiyonudur.

Tasarımcı bir sistemi tasarlariken, tasarımın ihtiyaç duyduğu donanım kaynağı yoğunluklarını tespit ederek buna uygun bir yonga seçer. Yonganın sahip olduğu donanım yoğunlu maliyete etki eden en önemli parametredir. FPGA üzerinde oluşturulacak olan bir YSA’nın boyutu, kullanılacak olan FPGA yongasının yoğunluğuna bağlıdır ve bu sınırlama tasarımcıyı daha küçük boyutlu ağlar oluşturmaya zorlamaktadır. Bununla birlikte IEEE 754 2008 32 bit kayan noktalı nümerik tanımlama yerine, 16 bit sabit noktalı nümerik tanımlama yöntemi kullanarak fazla miktarda kaynak kullanımına sahip aktivasyon fonksiyonu ve çarpıcı gibi elemanların boyutları küçültülerek, aynı yonga içerisinde daha büyük ölçekli YSA donanımları oluşturmak veya ilgili YSA’yı daha küçük/ucuz bir yonga üzerinde oluşturmak mümkündür. Altera Quartus II FPGA editörünün yonga içerisinde kullanılan donanım kaynaklarını gösteren Chip Planner eklentisinin FPANN ve FXANN donanımlarının FPGA yongasının ne kadarını kullandığı sırası ile Şekil 6.30 (a) ve (b)’de verilmiştir. FPANN ve FXANN donanımları Cyclone III EP3C120F780 yongası üzerinde gerçekleştirilmiştir [147].



Şekil 6.30. a) FPANN Chip Planner çıktısı, b) FXANN Chip Planner çıktısı.

FXANN tasarımı bloklar halinde verilmemiştir, çünkü bloklar FPANN tasarımında kullanılan bloklar ile aynı işlevselliğe sahiptir. Şematik gösterimlerindeki tek farklılık veri yollarının boyutudur. FPANN’de 32 bit veri yolları kullanılırken, FXANN’de 16 bit genişlikli veri yolları kullanılmıştır. Hafıza elemanları ve çarpım terimleri de aynı şekilde 32 bitten 16 bite düşmüştür. Ancak FXANN tasarımında, 16 bit sabit noktalı nümerik tanımlama ile gerçekleştirilen çarpım işlemlerinin sonuçları 32 bit çıktığı için bu sonuçlar 16 bite yuvarlanmıştır.

7. BÖLÜM

SONUÇ VE TARTIŞMA

Kardiyologlar hastalarını polikliniklerinde takip ederler. Ancak koroner kalp rahatsızlığı şikâyetleri ile hastanelere gelen hastaların çoğu ilk olarak acil servislere başvururlar. Kalp işaretlerinden aritmilerin tespit edilmesi özel bir eğitim ve tecrübe gerektirmektedir. Kardiyolog olmayan hekimler EKG verilerini yorumlamakta zorlanabilirler. Bu bakımlardan taşınabilir aritmi sınıflandırıcıları acil servislerde çalışan veya ambulanslardaki pratisyen hekimler tarafından, ön tanıya yardımcı olmaları açısından kullanılabilirler. Bazı tehlikeli kalp sorunları oluştuğunda bu sorunların ön tespiti ve ilk müdahalesi oldukça hayatidir. Ayrıca kardiyologun uzun süreli holter kayıtlarını takip etmesi ve sonuç çıkarması zaman alıcı, yorucu ve dikkat gerektiren bir işlemdir. Kardiyologlar otomatik aritmi sınıflandırıcı donanımını, uzun süreli holter kayıtlarından istatistikî veriler çıkarmak için de kullanabilirler.

Bununla birlikte kalpteki ritim bozukluklarının örüntüsü bazı durumlarda yanıltıcı olabilir. Çok bilinen bir aritmi örüntüsü bazen farklı bir aritmeyi ifade edebilir. Kalpteki morfolojik bozukluklar, alınan ilaçlar veya başka türlü sebepler EKG dalga şekillerinde bozulmalara sebep olabilir. Bu bakımdan hastaya tanısı konulurken, hikâyesinin alınması, fiziki muayenesinin yapılması ve laboratuvar tahlil sonuçlarının değerlendirilmesi gibi birçok ilave bilgi kullanılır. Otomatik aritmi sınıflandırıcılar, sadece doktorlara yardımcı olmaları amacını taşımaktadırlar.

Bilgisayar bağımlı algoritmalar, taşınabilir cihazlar kadar pratik bir kullanıma sahip olmadıkları için taşınabilir devre gerçeklemeleri klinikte büyük öneme sahiptirler. Otomatik aritmi sınıflandırıcı donanımları hem acil servislerde, hem ambulanslarda hem de sağlık ocaklarında, kardiyolog olmayan hekimlerce ön tanıya yardımcı bir araç olarak rahatlıkla kullanılabilirler.

Bu tez çalışmasında PVC tipi aritmileri otomatik olarak tanıyan sınıflandırıcı donanımları FPGA üzerinde donanımsal olarak gerçekleştirilmiştir. Sınıflandırıcı modeli olarak literatürde en yaygın olarak kullanılan teknik olan yapay sinir ağları seçilmiştir. Bununla birlikte literatürde önerilen ve medikal tanı sistemlerinde sıkça kullanılan YSA modellerinin neredeyse tamamı yazılımsal tabanlıdır. Yazılımsal YSA modelleri, işlemcilerin sıralı çalışma mantığı yüzünden biyolojik sinir sistemindeki paralel mimarinin avantajlarını sergileyemez ve gerçek zamanlı çalışmazlar. Buna ilave olarak, mevcut YSA modellerinin genellikle onlarca giriş ve gizli katman işlemci elemanından meydana geliyor olması, çıkış cevabının gecikmesinde ilave bir olumsuz etki olarak kendini göstermektedir. Biyolojik sinir ağlarındaki paralel mimariyi geleneksel mikroişlemciler veya DSP'ler ile oluşturmak mümkün değildir. Paralel mimari ancak ASIC ve FPGA yongaları veya analog devreler ile oluşturulabilirler. Ancak ASIC ve analog devrelerin sabit mimariye sahip olmaları ve ASIC yongalarının uzun tasarım süreçleri ve yüksek üretim maliyetleri büyük dezavantajdır. Bu yüzden gerek paralel mimariyi desteklemesi gerekse gerçek zamanlı çalışma avantajlarından dolayı, bu tez çalışmasında YSA donanım gerçeklemeleri, FPGA tasarım platformu kullanılarak yapılmıştır.

YSA içerisinde yer alan işlemci elemanlarda kullanılan aritmetik çarpıcı ve doğrusal olmayan aktivasyon fonksiyonu birimleri çok sayıda donanım kaynağı kullanırlar. Yazılımsal YSA modellerinde sıralı işlem mantığından dolayı çarpım ve diğer aritmetik işlemler, aynı donanımın sıralı olarak paylaşılmasıyla işletilmektedir. Bu sebepten YSA boyutunun büyüüp küçülmesi, kullanılan kaynak sayısını değil işlem sürelerini değiştirmektedir. Ancak literatürdeki mevcut yazılım tabanlı YSA modelleri paralel olarak FPGA üzerinde gerçekleştirilemeyecek kadar büyüktür. Giriş ve ara katmanlardaki işlemci elemanlarının sayısını azaltarak gerek eğitim sürelerini gerekse ağırlık hesaplama yükünü azaltmak amacı ile literatürde değişik birçok öznelik çıkarım/boyut azaltma algoritmaları kullanılmıştır. Bu yöntemlerden en popülerleri DD olmasına karşın bu yöntem genellikle YSA'lara göre daha karmaşık bir mimariye sahip olan bulanık yapay sinir ağlarında kullanılmışlardır. Bu tez çalışmasında boyut azaltma ve öznelik çıkarım yöntemi olarak, literatürde başarılı sonuçlar üreten ve birçok alanda da etkin bir şekilde kullanım alanı bulmuş TBA yöntemi seçilmiştir.

Donanım gerçeklemelerine geçilmeden önce, YSA tabanlı PVC sınıflandırıcıların yazılımsal modelleri Matlab ortamında tasarlanmıştır. Bu YSA modellerinden CGB, CGF, CGP, SCG, BPR ve LM eğitim algoritmaları ile eğitilmiş olan 544'er ağ modeli başarımları ve boyutları ile sırasıyla Tablo 5.2, Tablo 5.3, Tablo 5.4, Tablo 5.5, Tablo 5.6 ve Tablo 5.7'de verilmiştir. 544 farklı YSA konfigürasyonunun test hata oranları CGB için %3,54, CGF için %3,09, CGP için %3,73, SCG için %3,83, BPR için %2,53 ve LM için %2.17 değerindedir. Sonuçta LM algoritması ile daha yüksek sınıflandırma başarımları elde edildiği gözlemlenmiş ve donanım gerçeklemede kullanılacak referans ağ modelleri LM algoritması tarafından üretilmiştir. Tablo 5.7'ye bakıldığında gerek ortalama hata değerine yakın olması gerekse az sayıda giriş ve gizli katman işlemci elemanına sahip olmasından dolayı, sekiz temel bileşenin girişlerine uygulandığı ve gizli katmanında iki adet işlemsel eleman bulunan 8x2x1 boyutlu YSA modeli, donanım gerçekleştirme için referans olarak kabul edilmiştir. 8x2x1 boyutlu bu YSA modeli IEEE 32 bit kayan noktalı nümerik tanımlama ve 16 bit sabit noktalı nümerik tanımlama teknikleri ile ayrı ayrı tasarlanmış ve bu YSA donanımları sırası ile FPANN ve FXANN isimleri ile anılmıştır. FPANN ismi ile anılan YSA donanımı gizli katmandaki işlemci elemanlarda gerçek sigmoid aktivasyon fonksiyonu kullanırken, FXANN YSA donanımı gizli katmanda yer alan işlemci elemanlarda yedi parçalı doğrusal sigmoid aktivasyon fonksiyonu kullanılmıştır. Gerek kullanılan nümerik sayı formatları gerekse aktivasyon fonksiyonları FPANN ve FXANN donanım modellerinin sınıflandırma başarımlarına etki etmiştir.

Vargas ve arkadaşların yapmış oldukları bir çalışmada [16] daha büyük boyutlu bir YSA modelini, bu tez çalışmasında kullanılan kayıtlara (205, 208, 210 ve 215 numaralı MIT-BIH hasta kayıtları) TBA öznitelik çıkarım yöntemi uygulayarak eğitmiş olmalarına rağmen elde etmiş oldukları sınıflandırma başarımları bu tez çalışmasında elde edilen sınıflandırma başarımları kadar iyi değildir. Bunun sebebinin bu tez çalışmasında ham veri seti içerisindeki bozuk atımların çıkarılması olduğu düşünülmektedir. Çünkü kayıtlar içerisindeki bozuk veriler içerisinde anotasyon dosyası ile işaretlenmiş olan atım tipini temsil etmedikleri halde eğitim sonucunda üretilen ağırlıklara etki etmektedirler. Doğru sınıflandırılmış bir veri seti YSA'nın eğitiminde büyük bir etkiye sahiptir.

FPANN ile aynı ağırlık ve bias değerlerine sahip, 8x2x1 boyutlu bir Matlab Simulink YSA modeli oluşturulmuş ve bu modelin ürettiği sonuçlar FPANN donanımının ürettiği sonuçlar ile karşılaştırılmıştır. Sonuçta FPANN ve Simulink modellerinin ürettikleri cevaplar virgülden sonra beşinci haneye kadar aynı çıkmıştır. Bu derece yüksek hassasiyetin sebebi, FPANN’de kullanılan IEEE 754 32 bit kayan noktalı nümerik tanımlama ve gerçek sigmoid fonksiyonunun cevabından kaynaklanmaktadır. Öte yandan FXANN modeli ise kaynak kullanımını muazzam bir şekilde azalttığı için FPANN’dan çok daha kısa sürede sonuç üretmektedir. Ayrıca kaynak kullanımının azalması sistemin güç tüketimi ve maliyetlerini de düşürmüştür. IEEE 754 32 bit kayan noktalı nümerik tanımlama ile tasarlanan bir YSA yapısının, aynı ağ özellikleri ile daha düşük bir hassasiyetle, çok daha az yonga alanı kullanarak 16 bit nümerik tanımlama ile tasarlanması mümkündür. Tablo 6.10’da ifade edildiği gibi 8x2x1 boyutlu FXANN donanımı aynı boyutlu FPANN donanımından toplam lojik eleman sayısı bakımından 12,78, toplam DSP eleman sayısı bakımından 5,5 ve toplam hafıza elemanı bakımından 13,8 kat daha küçüktür. Şekil 6.30’da 8x2x1 boyutlu FPANN ve FXANN donanımlarının Cyclone III EP3C120F780 yongası üzerinde kullandıkları yonga alanları çarpıcı bir biçimde gösterilmiştir.

Taşınabilir donanım tasarımlarında doğruluk, maliyet, donanımın cevap süresi, güç tüketimi ve fiziksel boyutları önemli parametrelerdir. Bu bakımlardan sabit noktalı nümerik tanımlama taşınabilir cihaz teknolojilerinde kullanılmak için oldukça avantajlıdır. Çünkü sabit noktalı nümerik tanımlama standart binary aritmetiği kullandığı için daha az kaynak kullanır ve bu sebepten daha az güç tüketen, daha küçük boyutlu, daha düşük maliyetli ve daha kısa cevap sürelerine sahip donanımların oluşturulmasını mümkün kılar. Bu tez çalışmasında, 16 bit sabit noktalı nümerik tanımlama kullanılarak tasarlanmış olan FXANN, gizli katmanda iki, çıkış katmanında bir olmak üzere toplamda sadece üç işlemci eleman kullanarak, PVC tipi aritmlerin sınıflandırılmasında %96,54 başarımlı sağlamıştır.

Bununla birlikte nümerik hassasiyetin önemli olduğu uygulamalarda kayan noktalı nümerik tanımlama daha avantajlıdır. IEEE 754 32 bit kayan noktalı nümerik tanımlamanın kullanıldığı FPANN YSA donanımı %97,66 doğruluk ile FXANN’dan daha iyi bir sınıflandırma başarımlı sağlamıştır. Bu anlamda kaynak kullanımı ve nümerik hassasiyet arasında ters orantı mevcuttur. Problemin niteliğine göre ortaya

konulacak çözüm, tasarımcıyı nümerik hassasiyet veya kaynak kullanımı arasında bir seçim yapmaya itecektir.

Önerilen FXANN modelinin sınıflandırma başarımının arttırılması için giriş katmanına uygulanan öznitelik sayısı ve/veya gizli katmanda kullanılan işlemci elemanların sayısı arttırılabilir. 16 bit sabit noktalı nümerik tanımlama ile daha büyük ölçekli YSA modelleri, daha yüksek doğruluklarla sınıflandırma yapabilirler. Bu tez çalışmasının motivasyonu minimum donanım kaynakları ile elde edilebilecek en yüksek sınıflandırma başarımlarını yakalamak olduğu için daha büyük ölçekli ağların donanım modelleri oluşturulmamıştır. Ayrıca gerek 32 bit kayan noktalı gerekse 16 bit sabit noktalı nümerik tanımlamaların, YSA donanımlarının oluşturulmasında kullanılan temel aritmetik birimler için kaynak tüketimleri beşinci bölümde tablolar halinde verilmiştir. Bu tablolar kullanılarak Altera FPGA yongaları için her iki nümerik tanımlama yöntemine göre, istenilen büyüklükte bir ağ için gerekli donanım kaynağı hesaplanabilir.

FPANN ve FXANN donanımları FPGA üzerinde paralel olarak koşturulan yüksek hızlı sınıflandırıcı YSA donanımlarıdır. Bu donanımların EKG işareti gibi düşük frekanslı bir sinyal için gereğinden daha hızlı oldukları düşünülebilir. Fakat bu donanımlar günlük veya haftalık uzun holter kayıtlarının kardiyolog tarafından değerlendirilmesi için de kullanılabilir. Bu yüzden uzun süreli kayıtların incelenmesinde hızlı donanımların kullanılması büyük bir avantaj sağlamaktadır.

Bu tez çalışmasında tasarlanan YSA donanımları taşınabilir devre uygulamaları için yüksek bir doğrulukla donanım çözümleri sunmaktadır. FPANN ve FXANN donanımlarına giriş verileri üretmek için yardımcı bir işlemci veya bilgisayar tarafından gerçekleştirilen TBA öznitelik çıkarım işlemi FPGA içerisine alınarak bir analog ön çevirici maharetiyle, sayısallaştırılmış EKG verilerini işleyen bir tek yonga cihaz çözümü olabilir. Bir öznitelik çıkarım işlemi gerektirmeyen ve ham verilerin kullanıldığı birçok uygulamada, sayısallaştırılmış veriler direk olarak uygulanabilir. Gün içerisinde bir caddede oluşan trafik yoğunlukları veya hava tahmini gibi zaman veya sıcaklık gibi az sayıda parametrenin işlendiği problemler bu uygulamalara örnek olarak verilebilir.

Ayrıca bu tez çalışmasında önerilen donanım gerçeklemeleri, gerek biyomedikal işaretlerin işlenmesinde gerekse başka alanlarda kullanılan YSA modelleri için uygun bir donanım geliştirme platformu olarak kullanılabilir. Donanım çözümleri, YSA modellerinin sahada problemlere uygulanmasını mümkün kılmaları açısından oldukça önemlidir. Literatürde önerilen farklı disiplinleri de kapsayan çok sayıda YSA modeli bu yolla saha uygulamalarına aktarılabilir.

KAYNAKLAR

1. Karaolis, M. A., Moutiris, J. A., Hadjipanayi, D., Pattichis, C. S., Assessment of the Risk Factors of Coronary Heart Events Based on Data Mining With Decision Trees. *IEEE T Inf Technol B*, 14, 559-566, 2010.
2. WHO World Health Organization Cardiovascular Diseases. Available online: http://www.who.int/cardiovascular_diseases/en/, Ağustos, 2010.
3. Raphisak, P., Study of the Kalman Filter for Arrhythmia Detection with Intracardiac Electrograms, Yüksek Lisans Tezi, West Virginia University: Morgantown, US, 1999.
4. Kohler, B. U., Hennig, C., Orglmeister, R., The Principles of Software QRS Detection, *IEEE Eng Med Biol*, 21, 42-57, 2002.
5. de Chazal, P., O'Dwyer, M., Reilly, R. B., Automatic Classification of Heartbeats Using ECG Morphology and Heartbeat Interval Features, *IEEE T Bio-Med Eng*, 51, 1196-1206, 2004.
6. Romero, I., Grubb, N. R., Clegg, G. R., Robertson, C. E., Addison, P. S., Watson, J. N., T-Wave Alternans Found in Preventricular Tachyarrhythmias in CCU Patients Using a Wavelet Transform-Based Methodology, *IEEE T Bio-Med Eng*, 55, 2658-2665, 2008.
7. Sarkar, S., Ritscher, D., Mehra, R., A Detector for A Chronic Implantable Atrial Tachyarrhythmia Monitor, *IEEE T Bio-Med Eng*, 55, 1219-1224, 2008.
8. Zhang, X. S., Zhu, Y. S., Thakor, N. V., Wang, Z. Z., Detecting Ventricular Tachycardia and Fibrillation by Complexity Measure, *IEEE T Bio-Med Eng*, 46, 548-555, 1999.
9. Ripley, K. L., Bump, T. E., Arzbaeher, R. C., Evaluation of Techniques for Recognition of Ventricular Arrhythmias by Implanted Devices, *IEEE T Bio-Med Eng*, 36, 618-624, 1989.
10. Sayadi, O., Shamsollahi, M. B., Clifford, G. D., Robust Detection of Premature Ventricular Contractions Using a Wave-Based Bayesian Framework, *IEEE T Bio-Med Eng*, 57, 353-362, 2010.
11. Oresko, J. J., Jin, Z. P., Cheng, J., Huang, S. M., Sun, Y. W., Duschl, H., Cheng, A. C., A Wearable Smartphone-Based Platform for Real-Time Cardiovascular

- Disease Detection via Electrocardiogram Processing, *IEEE T Inf Technol B*, 14, 734-740, 2010.
12. Briggs, K. L., A Digital Approach to the Cardiac Cycle, *IEEE Eng Med Biol*, 13, 454-456, 1994.
 13. Trayanova, N., In the Spotlight: Cardiovascular Engineering, *IEEE Rev Biomed Eng*, 2, 12, 2009.
 14. Lim, J. S., Finding Features for Real-Time Premature Ventricular Contraction Detection Using a Fuzzy Neural Network System, *IEEE T Neural Networ*, 20, 522-527, 2009.
 15. İnan, O. T., Giovangrandi, L., Kovacs, G. T. A., Robust Neural-Network-Based Classification of Premature Ventricular Contractions using Wavelet Transform and Timing Interval Features, *IEEE T Bio-Med Eng*, 53, 2507-2515, 2006.
 16. Vargas, F., Lettnin, D., de Castro, M. C. F., Macarthy, M., Electrocardiogram Pattern Recognition by Means of MLP Network and PCA: A Case Study on Equal Amount of Input Signal Types, *VII Brazilian Symposium on Neural Networks, Proceedings*, 200-205, 2002.
 17. Ham, F. M., Han, S., Classification of Cardiac Arrhythmias Using Fuzzy ARTMAP, *IEEE T Bio-Med Eng*, 43, 425-430, 1996.
 18. Jiang, W., Kong, S. G., Block-Based Neural Networks for Personalized ECG Signal Classification, *IEEE T Neural Networ*, 18, 1750-1761, 2007.
 19. Shyu, L. Y., Wu, Y. H., Hu, W. C., Using Wavelet Transform and Fuzzy Neural Network for VPC Detection from the Holter ECG, *IEEE T Bio-Med Eng*, 51, 1269-1273, 2004.
 20. Özbay, Y., Karlik, B., A Recognition of ECG Arrhythmias using Artificial Neural Networks, *23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1(4), 23, 1680-1683, 2001.
 21. Özdemir, A. T., Danışman, K., Asyalı, M. H., FPGA Based Arrhythmia Classifier. *Biyomut: 14th National Biomedical Engineering Meeting*, 17-20, 2009.
 22. Gacek, A., Pedrycz, W., A Genetic Segmentation of ECG Signals, *IEEE T Bio-Med Eng*, 50, 1203-1208, 2003.
 23. Andreao, R. V., Dorizzi, B., Boudy, J., ECG Signal Analysis Through Hidden Markov Models, *IEEE T Bio-Med Eng*, 53, 1541-1549, 2006.

24. Murthy, I. S., Rangaraj, M. R., New Concepts for PVC Detection, *IEEE Trans Biomed Eng*, 26, 409-416, 1979.
25. Thakor, N. V., Zhu, Y. S., Applications of Adaptive Filtering to ECG Analysis - Noise Cancellation and Arrhythmia Detection, *IEEE T Bio-Med Eng*, 38, 785-794, 1991.
26. Rojo-Alvarez, J. L., Barquero-Perez, O., Mora-Jimenez, I., Everss, E., Rodriguez-Gonzalez, A. B., Garcia-Alberola, A., Heart Rate Turbulence Denoising Using Support Vector Machines, *IEEE T Bio-Med Eng*, 56, 310-319, 2009.
27. Lin, K. P., Chang, W. H., QRS Feature-Extraction Using Linear Prediction, *IEEE T Bio-Med Eng*, 36, 1050-1055, 1989.
28. Hallstrom, A. P., Stein, P. K., Schneider, R., Hodges, M., Schmidt, G., Ulm, K., Structural Relationships Between Measures Based on Heart Beat Intervals: Potential for Improved Risk Assessment, *IEEE T Bio-Med Eng*, 51, 1414-1420, 2004.
29. Reddy, B. R. S., Murthy, I. S. N., Chatterjee, P. C., Rhythm Analysis using Vectorcardiograms, *IEEE T Bio-Med Eng*, 32, 97-104, 1985.
30. Senhadji, L., Carrault, G., Bellanger, J. J., Passariello, G., Comparing Wavelet Transforms for Recognizing Cardiac Patterns, *IEEE Eng Med Biol*, 14, 167-173, 1995.
31. Kadambe, S., Murray, R., Boudreaux-Bartels, G. F., Wavelet Transform-Based QRS Complex Detector, *IEEE Trans Biomed Eng*, 46, 838-848, 1999.
32. Kobayashi, K., Uchikawa, Y., Frequency Analysis of Premature Ventricular Contraction Using 3-D MCG Measurements, *IEEE T Magn*, 35, 4112-4114, 1999.
33. West, D., West, V., Improving Diagnostic Accuracy Using a Hierarchical Neural Network to Model Decision Subtasks, *Int J Med Inform*, 57, 41-55, 2000.
34. Hu, Y. H., Tompkins, W. J., Urrusti, J. L., Afonso, V. X., Applications of Artificial Neural Networks for ECG Signal-Detection and Classification, *J Electrocardiol*, 26, 66-73, 1993.
35. MIT-BIH Arrhythmia Database Harvard-MIT Division of Health Sciences and Technology, <http://ecg.mit.edu/>, Austos, 2010.

36. Dokur, Z., Ölmez, T., Yazgan, E., Comparison of Discrete Wavelet and Fourier Transforms for ECG Beat Classification, *Electron Lett*, 35, 1502-4, 1999.
37. Mayr, C., Ehrlich, M., Henker, S., Wendt, K., Schuffny, R., Mapping Complex, Large - Scale Spiking Networks on Neural VLSI, *Proc Wrlld Acad Sci E*, 19, 40-45, 2007.
38. Dinu, A., Cirstea, M. N., Cirstea, S. E., Direct Neural-Network Hardware-Implementation Algorithm, *IEEE T Ind Electron*, 57, 1845-1848, 2010.
39. Medical Picture Library, King Saud University, <http://www.ksu.edu.sa>, Ağustos, 2010.
40. Ferry, D. R., *Basic Electrocardiography in Ten Days*, McGraw-Hill: New York, London 2001.
41. Guyton, A. C., Hall, J. E., *Textbook of Medical Physiology*, 10. Baskı, Saunders: Philadelphia 2000.
42. Clifford, G. D., Azuaje, F., McSharry, P. E., *Advanced Methods and Tools for ECG Data Analysis*, Artech House: Boston, Mass., London 2006.
43. Jenkis, D., A Brief History of ECG, <http://www.ecglibrary.com/ecghist.html>, Ağustos, 2010.
44. MIT-OCW HST.582J / 6.555J / 16.456J Biomedical Signal and Image Processing, <http://ocw.mit.edu>, Ağustos, 2010.
45. Marriott, H. J. L., Schwartz, N. L., Bix, H. H., Ventricular Fusion Beats. *Circulation*, 26, 880-884, 1962.
46. AAMI Association for the Advancement of Medical Instrumentation, <http://www.aami.org/>, Ağustos, 2010.
47. PhysioNet, The Research Resource for Complex Physiologic Signals, <http://www.physionet.org>, Ağustos, 2010.
48. Cygwin Linux-like environment for Windows, <http://www.cygwin.com/>, Ağustos, 2010.
49. Kyoso, M., Uchiyama, A., Development of an ECG Identification System, *Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1(4), 23, 3721-3723, 2001.
50. Draper, H. W., Peffer, C. J., Stallmann, F. W., Littmann, D., Pipberger, H. V., The Corrected Orthogonal Electrocardiogram and Vectorcardiogram in 510 Normal Men (Frank Lead System), *Circulation*, 30, 853-864, 1964.

51. Kozmann, G., Lux, R. L., Green, L. S., Sources of Variability in Normal Body Surface Potential Maps, *Circulation*, 79, 1077-1083, 1989.
52. Poree, F., Bansard, J. Y., Kervio, G., Carrault, G., Stability Analysis of the 12-Lead ECG Morphology in Different Physiological Conditions of Interest for Biometric Applications, *Cinc: 2009 36th Annual Computers in Cardiology Conference*, 285-288, 2009.
53. Chan, A. D. C., Hamdy, M. M., Badre, A., Badee, V., Wavelet Distance Measure for Person Identification Using Electrocardiograms, *IEEE T Instrum Meas*, 57, 248-253, 2008.
54. Rao, K. D., DWT Based Detection of R-Peaks and Data Compression of ECG Signals. *Iete J Res*, 43, 345-349, 1997.
55. Fernandez, J., Harris, M., Meyer, C., Combining Algorithms in Automatic Detection of R-Peaks in ECG Signals, *18th IEEE Symposium on Computer-Based Medical Systems, Proceedings*, 297-302, 2005.
56. Chagas, A. V., Da Silva, E. A. B., Nadal, J., ECG Data Compression Using Wavelets, *Computers in Cardiology 2000*, 27, 423-426, 2000.
57. Alesanco, A., Olmos, S., Istepanian, R., Garcia, J., A Novel Real-Time Multilead ECG Compression and de-Noising Method Based on the Wavelet Transform, *Comput Cardiol*, 30, 593-596, 2003.
58. de Chazal, P., Reilly, R. B., A Patient-Adapting Heartbeat Classifier Using ECG Morphology and Heartbeat Interval Features, *IEEE T Bio-Med Eng*, 53, 2535-2543, 2006.
59. Risk, M. R., Sobh, J. F., Saul, J. P., Beat Detection and Classification of ECG Using Self Organizing Maps, *Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 19, 89-91, 1997.
60. Bulusu, S., Detection of ECG Transient ST-Segment Episodes and Machine Learning Based Heart Beat Classification, *Yüsek Lisans Tezi*, University of Texas, Dallas, 2010.
61. Polikar, R., The Wavelet Tutorial,
<http://users.rowan.edu/~polikar/wavelets/WTutorial.html>, Agosto, 2010.

62. Langley, P., Bowers, E. J., Murray, A., Principal Component Analysis as a Tool for Analyzing Beat-to-Beat Changes in ECG Features: Application to ECG-Derived Respiration, *IEEE T Bio-Med Eng*, 57, 821-829, 2010.
63. Jain, A. K., Mao, J. C., Mohiuddin, K. M., Artificial Neural Networks: A Tutorial, *Computer*, 29, 31-+, 1996.
64. Jolliffe, I. T., Principal Component Analysis, 2. Baskı, Springer: NewYork 2002.
65. Ersoy, N., Önemli Bileşenler Analizinin Kuramı ve Bir Uygulama, 1. Baskı, Ankara İktisadi ve Ticari İlimler Akademisi: Ankara 1981.
66. Cady, L. D., Jr., Woodbury, M. A., Tick, L. J., Gertler, M. M., A Method for Electrocardiogram Wave-Pattern Estimation Example: Left Ventricular Hypertrophy, *Circ Res*, 9, 1078-1082, 1961.
67. Yasui, S., Whipple, G. H., Stark, L., Comparison of Human and Computer Electrocardiographic Wave-Form Classification and Identification, *American Heart J*, 68, 236-242, 1964.
68. Vijaya, G., Kumar, V., Verma, H. K., ANN-Based QRS-Complex Analysis of ECG, *J Med Eng Technol*, 22, 160-167, 1998.
69. Strintzis, M. G., Maglaveras, N., Stalidis, G., Magnisalis, X., Use of Neural Networks for Electrocardiogram Feature-Extraction, Recognition and Classification, 93, 50-54, 1993.
70. Xue, Q. Z., Hu, Y. H., Tompkins, W. J., Neural-Network-Based Adaptive Matched Filtering for QRS Detection, *IEEE T Bio-Med Eng*, 39, 317-329, 1992.
71. Atoui, H., Fayn, J., Rubel, P., A Novel Neural-Network Model for Deriving Standard 12-Lead ECGs from Serial Three-Lead ECGs: Application to Self-Care, *IEEE T Inf Technol B*, 14, 883-890, 2010.
72. Poli, R., Cagnoni, S., Valli, G., Genetic Design of Optimum Linear and Nonlinear QRS Detectors, *IEEE T Bio-Med Eng*, 42, 1137-1141, 1995.
73. Afonso, V. X., Tompkins, W. J., Nguyen, T. Q., Luo, S., ECG Beat Detection Using Filter Banks, *IEEE T Bio-Med Eng*, 46, 192-202, 1999.
74. Daskalov, I. K., Christov, I. I., Electrocardiogram Signal Preprocessing for Automatic Detection of QRS Boundaries, *Med Eng Phys*, 21, 37-44, 1999.
75. Ebenezer, D., Papa, F. F., A Recursive Digital Differentiator for ECG Preprocessing, *Med Eng Phys*, 16, 273-277, 1994.

76. de Pinto, V., Filters for the Reduction of Baseline Wander and Muscle Artifact in the ECG, *J Electrocardiol*, 25, 40-48, 1992.
77. Suppappola, S., Sun, Y., Nonlinear Transforms of ECG Signals for Digital QRS Detection: A Quantitative Analysis, *IEEE T Bio-Med Eng*, 41, 397-400, 1994.
78. Trahanias, P. E., An Approach to QRS Complex Detection Using Mathematical Morphology, *IEEE T Bio-Med Eng*, 40, 201-205, 1993.
79. Orr, G., Schraudolph, N., Cummins, F., Computation in the Brain, <http://www.willamette.edu/~gorr/classes/cs449/brain.html>, Augustos 2010.
80. Colla, A. M., Parenti, R., Neural Networks Applications in Process Control, *Prog Neural Process*, 8, 43-54, 1997.
81. Marengo, E., Robotti, E., Bobba, M., Liparota, M. C., Artificial Neural Networks Applications in the Field of Separation Science Optimisation, *Curr Anal Chem*, 2, 181-194, 2006.
82. John, S., A Convergence Instability Analysis of Neural Networks Applications in Financial Data Sets, *International Conference on Artificial Intelligence*, 1(2), 451-457, 1999.
83. Udo, G. J., Neural Networks Applications in Manufacturing Processes, *Comput Ind Eng*, 23, 97-100, 1992.
84. McCulloch, W. S., Pitts, W., A Logical Calculus of the Ideas Immanent in Nervous Activity, *Bull Math Biol*, 52, 173-197, 1990.
85. Hebb, D. O., *The organization of behavior: A Neuropsychological Theory*, Wiley: New York, 1949.
86. Rosenblatt, F., The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, *Psychol Rev*, 65, 386-408, 1958.
87. Widrow, B., Hoff, M. E., Adaptive Switching Circuits, In *Neurocomputing: Foundations of Research*, MIT Press: 1988.
88. Daqi, Z., The Research Progress and Prospects of Artificial Neural Networks, *Journal of Southern Yangtze University (Natural Science Edition)*, 103-110, 2004.
89. Minsky, M. L., Papert, S., *Perceptrons: An Introduction to Computational Geometry*, MIT Press: Cambridge, Mass., 1969.
90. Hopfield, J. J., Neural Networks and Physical Systems with Emergent Collective Computational Abilities, *P Natl Acad Sci-Biol*, 79, 2554-2558, 1982.

91. Hopfield, J. J., Learning Algorithms and Probability Distributions in Feed-Forward and Feed-Back Networks, *Proc Natl Acad Sci U S A*, 84, 8429-8433, 1987.
92. Kohonen, T., *Self-Organization and Associative Memory*, Springer-Verlag: 1984.
93. Grossberg, S., Stone, G., Neural Dynamics of Word Recognition and Recall - Attentional Priming, Learning, and Resonance, *Psychological Review*, 93, 46-74, 1986.
94. McClelland, J. L., Rumelhart, D. E., *Explorations in Parallel Distributed Processing : A Handbook of Models, Programs and Exercises*, MIT Press: Cambridge, Mass. 1988.
95. Liu, J. H., Liang, D. Q., A Survey of FPGA-Based Hardware Implementation of ANNs, *International Conference on Neural Networks and Brain*, 1(3), 915-918, 2005.
96. Hodgkin, A. L., Huxley, A. F., A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in Nerve, *J Physiol*, 117, 500-544, 1952.
97. Liao, Y. J., Safa, P., Chen, Y. R., Sobel, R. A., Boyden, E. S., Tsien, R. W., Anti-Ca²⁺ Channel Antibody Attenuates Ca²⁺ Currents and Mimics Cerebellar Ataxia In-Vivo, *Proc Natl Acad Sci U S A*, 105, 2705-2710, 2008.
98. Dagtekin, M., Sinir Hücreleri ile İletişim, *Bilişim Dergisi*, 3, 1-2, 2009.
99. Alavala, C. R., Ebrary Inc., *Fuzzy Logic and Neural Networks Basic Concepts and Application*, New Age International (P) Ltd., Publishers: New Delhi.
100. Kim, B., Hong, W. S., Use of Neural Network to Characterize a Low Pressure Temperature Effect on Refractive Property of Silicon Nitride Film Deposited by PECVD, *IEEE T Plasma Sci*, 32, 84-89, 2004.
101. Merchant, S., Peterson, G., Kong, S., Intrinsic Embedded Hardware Evolution of Block-Based Neural Networks, *International Conference on Engineering of Reconfigurable Systems & Algorithms*, 2006
102. Rumelhart, D. E., A Parallel Distributed-Processing Account of the Acquisition of Past Tense in English, *B Psychonomic Soc*, 24, 326-326, 1986.
103. Dorf, R. C., *The Electrical Engineering Handbook*, 3. Baskı, CRC/Taylor & Francis: Boca Raton 2006.

104. Graupe, D., Principles of Artificial Neural Networks, World Scientific: Singapore: River Edge, NJ 1997.
105. Alavala, C. R., Fuzzy Logic and Neural Networks : Basic Concepts and Application, New Age International: Daryaganj, Delhi, IND 2008.
106. Haykin, S. S., Neural networks : A Comprehensive Foundation, Maxwell Macmillan International: New York 1994.
107. Zurada, J. M., Introduction to Artificial Neural Systems, West: St. Paul 1992.
108. Graves, A., Liwicki, M., Fernandez, S., Bertolami, R., Bunke, H., Schmidhuber, J., A Novel Connectionist System for Unconstrained Handwriting Recognition, IEEE T Pattern Analysis and Machine Intelligence, 31, 855-868, 2009.
109. Rumelhart, D. E., McClelland, J. L., University of California San Diego. PDP Research Group., Parallel Distributed Processing : Explorations in the Microstructure of Cognition, MIT Press: Cambridge, Mass. 1986.
110. Carpenter, G. A., Grossberg, S., The Art of Adaptive Pattern-Recognition by a Self-Organizing Neural Network, Computer, 21, 77-88, 1988.
111. Hinton, G. E., Sejnowski, T. J., Unsupervised Learning : Foundations of Neural Computation, MIT Press: Cambridge, Mass. 1999.
112. Ackley, D. H., Hinton, G. E., Sejnowski, T. J., A Learning Algorithm for Boltzmann Machines, Cognitive Sci, 9, 147-169, 1985.
113. Fine, T. L., Feedforward Neural Network Methodology, Springer: New York 1999.
114. Beale, M. H., Hagan, M. T., Demuth, H. B. Neural Network Toolbox™ 7, User's Guide, http://www.mathworks.com/help/pdf_doc/nnet/nnet.pdf, Ağustos 2010.
115. Man, Z. H., Wu, H. R., Liu, S., Yu, X. H., A New Adaptive Backpropagation Algorithm Based on Lyapunov stability Theory for Neural Networks, IEEE T Neural Networ, 17, 1580-1591, 2006.
116. Bishop, C. M., Neural Networks for Pattern Recognition, Clarendon Press; Oxford University Press: Oxford, New York 1995.
117. Nawa, N. E., Korkin, M., de Garis, H., ATR's CAM-Brain Project: The Evolution of Large-Scale Recurrent Neural Network Modules, International Conference on Parallel and Distributed Processing Techniques and Applications, 1(4), 1087-1094, 1998.

118. Asyalı, M. H., Özdemir, A. T., Aksebzeci, B. H., K., Danışman, Aritmi Sınıflandırmada Kullanılan Yapay Sinir Ağlarının Karmaşıklığın Azaltılması, BİYOMUT 2008, ODTÜ, Ankara, Türkiye, 2008.
119. Altera Altera Inc., The Programmable Logic Company, www.altera.com, Ağustos, 2010.
120. Wakerly, J. F., Digital Design Principles and Practices, 3. Baskı, Prentice Hall: Upper Saddle River, NJ 1999.
121. Ashour, M. A., Saleh, H. I., An FPGA Implementation Guide for Some Different Types of Serial-Parallel Multiplier Structures, Microelectr J, 31, 161-168, 2000.
122. Chen, W.-K., The VLSI handbook, 2. Baskı, CRC/Taylor & Francis: Boca Raton, FL 2007.
123. Navabi, Z., Digital Design and Implementation with Field Programmable Devices, Kluwer Academic Publishers: Norwell 2005.
124. Pedroni, V. A., Circuit Design with VHDL, MIT Press: Cambridge, Mass. 2004.
125. Kuon, I., Rose, J., Measuring the Gap Between FPGAs and ASICs, IEEE T Comput Aid D, 26, 203-215, 2007.
126. Pandya, V., A Handel-C Implementation of the Back-Propagation Algorithm on Field Programmable Gate Arrays, Yüksek Lisans Tezi, University of Guelph: 2006.
127. Allen, R., Compiling High-Level Languages to DSPs, IEEE Signal Proc Mag, 22, 47-56, 2005.
128. Lee, D. U., Luk, W., Villasenor, J. D., Zhang, G. L., Leong, P. H. W., A Hardware Gaussian Noise Generator Using the Wallace Method, IEEE T VLSI Syst, 13, 911-920, 2005.
129. Roy, S., Banerjee, P., An Algorithm for Trading off Quantization Error with Hardware Resources for MATLAB-Based FPGA Design, IEEE T Comput, 54, 886-896, 2005.
130. Navabi, Z., VHDL : Analysis and Modeling of Digital Systems, 2. Baskı, McGraw-Hill: New York 1997.
131. Navabi, Z., Embedded Core Design with FPGAs, McGraw-Hill: New York ; London 2007.

132. Do, G. L., Feher, K., Efficient Filter Design for IS-95 CDMA Systems, *IEEE T Consum Electr*, 42, 1011-1020, 1996.
133. Kurdthongmee, W., Design and Implementation of an FPGA-Based Multiple-Colour LED Display Board, *Microprocess Microsy*, 29, 327-336, 2005.
134. Salcic, Z., Cao, J. Y., Nguang, S. K., A Floating-Point FPGA-Based Self-Tuning Regulator, *IEEE T Ind Electron*, 53, 693-704, 2006.
135. Wang, L., Johannessen, E. A., Hammond, P. A., Cui, L., Reid, S. W. J., Cooper, J. M., Cumming, D. R. S., A Programmable Microsystem Using System-on-Chip for Real-Time Biotelemetry, *IEEE T Bio-Med Eng*, 52, 1251-1260, 2005.
136. Krupnova, H., Dinh, D. A. V., Saucier, G., Boubal, M., Real Time Prototyping Method and a Case Study, *P IEEE Rap Syst Prot*, 13-18, 1998.
137. Plante, J., Shaw, H., Mickens, L., Johnson-Bey, C., Overview of Field Programmable Analog Arrays as Enabling Technology for Evolvable Hardware for High Reliability Systems, *Nasa/Dod Conference on Evolvable Hardware*, 77-78, 2003.
138. Bade, S. L., Hutchings, B. L., In FPGA-Based Stochastic Neural Networks-Implementation, *IEEE Workshop FPGAs for Custom Computing Machines*, 189-198, 1994.
139. Hikawa, H., An Efficient Three-Valued Multilayer Neural Network with On-Chip Learning Suitable for Hardware Implementation, *Systems and Computers in Japan*, 31, 43-51, 2000.
140. Hikawa, H., A Multilayer Neural Network with Pulse Position Modulation. *Systems and Computers in Japan*, 34, 36-46, 2003.
141. Schoenauer, T., Jahnke, A., Roth, U., Klar, H., Digital Neurohardware: Principles and Perspectives, *Anwendung NN'98*, 101-106, Magdenburg, Germany, 1998.
142. Hammerstrom, D., A Highly Parallel Digital Architecture for Neural Network Emulation, *VLSI Artificial Intelligence and Neural Networks*, 357-366, 1991.
143. Ramacher, U., Synapse - a Neurocomputer that Synthesizes Neural Algorithms on a Parallel Systolic Engine, *J Parallel Distr Com*, 14, 306-318, 1992.
144. Heemskerk, J., Overview of Neural Hardware in Neurocomputers for Brain-Style Processing, *Doktora Tezi*, Leiden University, 1995.

145. Nestor Inc., Ni1000 Recognition Accelerator,
<http://www.warthman.com/images/Ni1000ds.pdf>, Ağustos, 2010.
146. de Garis, H., Gers, F., Korkin, M., Agah, A., Nawa, N., ATR's Artificial Brain (CAM-Brain) - Project a Progress Report, IEEE International Conference on Evolutionary Computation, 51-56, 1998.
147. Özdemir, A. T., Danışman, K., Fully Parallel ANN-Based Arrhythmia Classifier on a Single Chip FPGA: FPAAC, Turkish Journal of Electric, Yayınlanmak Üzere Kabul Edildi, 2010.
148. Ifrah, G., A Universal History of Computing : from the Abacus to the Quantum Computer, John Wiley: New York 2000.
149. Richard, D., Everything Stems From Nothing, 1. Baskı, Clemson University: 2007.
150. Wikipedia History of the Hindu–Arabic Numeral System,
http://en.wikipedia.org/wiki/History_of_the_Hindu%E2%80%93Arabic_numeral_system, 2010.
151. Erle, M. A., Algorithms and Hardware Designs for Decimal Multiplication, Lehigh University: 2008.
152. Goldstine, H. H., Goldstine, A., The Electronic Numerical Integrator and Computer (ENIAC), IEEE Annals of the History of Computing, 18, 10-16, 1996.
153. Head, R., Univac: A Philadelphia Story, IEEE Annals of the History of Computing, 23, 60-63, 2001.
154. Knuth, D. E., The IBM 650: An Appreciation from the Field, IEEE Annals of the History of Computing, 8, 50-55, 1986.
155. Stern, N., From ENIAC to UNIVAC an Appraisal of the Eckert-Mauchly Computers, Bedford, MA: Digital Press 1981.
156. Wilkes, M. V., Arithmetic on the EDSAC, IEEE Annals of the History of Computing, 19, 13-15, 1997.
157. Williams, M. R., The Origins, Uses, and Fate of the EDVAC, IEEE Annals of the History of Computing, 15, 22-38, 1993.
158. Leser, T., Romanelli, M., Programming and Coding for ORDVAC,
<http://www.bitsavers.org/pdf/ordvac/-ORDVAC-programming-Oct56.pdf>, Ağustos, 2010.

159. Gray, G., Unisys History Newsletter,
<http://www.cc.gatech.edu/gvu/people/randy.carpenter/folklore/v1n2.html>,
Ağustos, 2010.
160. Saleh, H. H. M., Fused Floating-Point Arithmetic for DSP, University of Texas,
Austin: 2009.
161. IEEE, Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Standard 754-
1985, 1985.
162. IEEE, Standard for Floating-Point Arithmetic, IEEE Standard 754-2008, 2008.
163. Holt, J. L., Baker, T. E. In Back Propagation Simulations Using Limited
Precision Calculations, Neural Networks, IJCNN-91-Seattle International Joint
Conference, 122, 121-126, 1991.
164. Nichols, K. R., Moussa, M. A., Areibi, S. M., Feasibility of Floating-Point
Arithmetic in FPGA Based ANNs, Computer Applications in Industry and
Engineering, 8-13, 2002.

EKLER

A. 544 Değişik Boyutlu Ağın Eğitiminde Kullanılan Matlab Kodları

```

close all, clear all
ann205 = readannot('205.atr');
ann208 = readannot('208.atr');
ann210 = readannot('210.atr');
ann213 = readannot('213.atr');
save aritmi ann205 ann208 ann210 ann213

F_index1 = find(ann205.annntyp == 'F'); L1 = length(F_index1);
F_index2 = find(ann208.annntyp == 'F'); L2 = length(F_index2);
F_index3 = find(ann210.annntyp == 'F'); L3 = length(F_index3);
F_index4 = find(ann213.annntyp == 'F'); L4 = length(F_index4);

F_class_data1 = zeros(L1,181);
for i = 1:L1
    t_marker = ann205.time(F_index1(i));
    ecg_data = rdsign212('205.dat',2,t_marker-90,t_marker+90);
    F_class_data1(i,:) = ecg_data(1,:);
end
F_class_data1_good = inconsistent_remover(F_class_data1);

F_class_data2 = zeros(L2,181);
for i = 1:L2
    t_marker = ann208.time(F_index2(i));
    ecg_data = rdsign212('208.dat',2,t_marker-90,t_marker+90);
    F_class_data2(i,:) = ecg_data(1,:);
end
F_class_data2_good = inconsistent_remover(F_class_data2);

F_class_data3 = zeros(L3,181);
for i = 1:L3
    t_marker = ann210.time(F_index3(i));
    ecg_data = rdsign212('210.dat',2,t_marker-90,t_marker+90);
    F_class_data3(i,:) = ecg_data(1,:);
end
F_class_data3_good = inconsistent_remover(F_class_data3);

F_class_data4 = zeros(L4,181);
for i = 1:L4
    t_marker = ann213.time(F_index4(i));
    ecg_data = rdsign212('213.dat',2,t_marker-90,t_marker+90);
    F_class_data4(i,:) = ecg_data(1,:);
end
F_class_data4_good = inconsistent_remover(F_class_data4);

```

```

% figure(1)
% subplot(221), plot(F_class_data1_good')
% subplot(222), plot(F_class_data2_good')
% subplot(223), plot(F_class_data3_good')
% subplot(224), plot(F_class_data4_good')

V_index1 = find(ann205.anntyp == 'V'); L1 = length(V_index1);
V_index2 = find(ann208.anntyp == 'V'); L2 = length(V_index2);
V_index3 = find(ann210.anntyp == 'V'); L3 = length(V_index3);
V_index4 = find(ann213.anntyp == 'V'); L4 = length(V_index4);

V_class_data1 = zeros(L1,181);
for i = 1:L1
    t_marker = ann205.time(V_index1(i));
    ecg_data = rdsign212('205.dat',2,t_marker-90,t_marker+90)';
    V_class_data1(i,:) = ecg_data(1,:);
end
V_class_data1_good = inconsistent_remove(V_class_data1);

V_class_data2 = zeros(L2,181);
for i = 1:L2
    t_marker = ann208.time(V_index2(i));
    ecg_data = rdsign212('208.dat',2,t_marker-90,t_marker+90)';
    V_class_data2(i,:) = ecg_data(1,:);
end
V_class_data2_good = inconsistent_remove(V_class_data2);

V_class_data3 = zeros(L3,181);
for i = 1:L3
    t_marker = ann210.time(V_index3(i));
    ecg_data = rdsign212('210.dat',2,t_marker-90,t_marker+90)';
    V_class_data3(i,:) = ecg_data(1,:);
end
V_class_data3_good = inconsistent_remove(V_class_data3);

V_class_data4 = zeros(L4,181);
for i = 1:L4
    t_marker = ann213.time(V_index4(i));
    ecg_data = rdsign212('213.dat',2,t_marker-90,t_marker+90)';
    V_class_data4(i,:) = ecg_data(1,:);
end
V_class_data4_good = inconsistent_remove(V_class_data4);

% figure(2)
% subplot(221), plot(V_class_data1_good')
% subplot(222), plot(V_class_data2_good')
% subplot(223), plot(V_class_data3_good')
% subplot(224), plot(V_class_data4_good')

```

```

N_index1 = find(ann205.anntyp == 'N'); L1 = length(N_index1);
N_index2 = find(ann208.anntyp == 'N'); L2 = length(N_index2);
N_index3 = find(ann210.anntyp == 'N'); L3 = length(N_index3);
N_index4 = find(ann213.anntyp == 'N'); L4 = length(N_index4);

N_class_data1 = zeros(L1,181);
for i = 1:L1
    t_marker = ann205.time(N_index1(i));
    ecg_data = rdsign212('205.dat',2,t_marker-90,t_marker+90);
    N_class_data1(i,:) = ecg_data(1,:);
end
N_class_data1_good = inconsistent_removal(N_class_data1);

N_class_data2 = zeros(L2,181);
for i = 1:L2
    t_marker = ann208.time(N_index2(i));
    ecg_data = rdsign212('208.dat',2,t_marker-90,t_marker+90);
    N_class_data2(i,:) = ecg_data(1,:);
end
N_class_data2_good = inconsistent_removal(N_class_data2);

N_class_data3 = zeros(L3,181);
for i = 1:L3
    t_marker = ann210.time(N_index3(i));
    ecg_data = rdsign212('210.dat',2,t_marker-90,t_marker+90);
    N_class_data3(i,:) = ecg_data(1,:);
end
N_class_data3_good = inconsistent_removal(N_class_data3);

N_class_data4 = zeros(L4,181);
for i = 1:L4
    t_marker = ann213.time(N_index4(i));
    ecg_data = rdsign212('213.dat',2,t_marker-90,t_marker+90);
    N_class_data4(i,:) = ecg_data(1,:);
end
N_class_data4_good = inconsistent_removal(N_class_data4);

% figure(3)
% subplot(221), plot(N_class_data1_good)
% subplot(222), plot(N_class_data2_good)
% subplot(223), plot(N_class_data3_good)
% subplot(224), plot(N_class_data4_good)

```

```

F_class =
[F_class_data1_good;F_class_data2_good;F_class_data3_good;F_class_data4_good];
L1 = size(F_class,1);

V_class =
[V_class_data1_good;V_class_data2_good;V_class_data3_good;V_class_data4_good];
L2 = size(V_class,1);

N_class =
[N_class_data1_good;N_class_data2_good;N_class_data3_good;N_class_data4_good];
L3 = size(N_class,1);

N_class = N_class(randperm(round(L3/5)),:); L3 = size(N_class,1);

all_data = [F_class;V_class;N_class];
new_data = all_data;

for i = 1:size(all_data,2)
    col_data = all_data(:,i);
    mean_col = mean(col_data);
    std_col = std(col_data);
    new_data(:,i) = (col_data - mean_col)/(5*std_col);
end

[prin_comps,new2_data,latent,tsquare] = princomp(new_data);

i=2;
j=5;
for i = 2:35
    for j=5:20
        P = new2_data(:,1:j)';
        T = [ones(L1,1);2*ones(L2,1);3*ones(L3,1)]';

net = newff(P,T,j,{'logsig','purelin'});
% net.performFcn='hata algoritması';
% net.trainFcn = 'öğrenme algoritması';
net.trainParam.goal = 1e-6;
net.trainParam.show = 20;
net.trainParam.epochs = 1500;
% net.trainParam.mc = 0.95;
net.trainParam.max_fail = 20;
% net.trainParam.min_grad = eps;
[net,tr] = train(net,P,T);

```

```

a = sim(net,P);
b = a;
b(a<1.5) = 1;
b(a>=1.5 & a<2.5) = 2;
b(a>=2.5) = 3;
%figure
%plot(T)
%hold on
%plot(b,'r+')
Percent_err = 100*sum(b~=T)/length(T);
i,j,Percent_err,
report_NN(i,j)= Percent_err;
    end
end

report=report_NN(:,1)

```

B. Parçalı Doğrusal Sigmoid Aktivasyon Fonksiyonunun Matlab Kodları

```

function out1 = sigpw3(in1,in2,in3,in4)
%LOGSIG Logarithmic sigmoid transfer function.
%
% Syntax
%
%   A = logsig(N,FP)
%   dA_dN = logsig('dn',N,A,FP)
%   INFO = logsig(CODE)
%
% Description
%   LOGSIG(N,FP) takes N and optional function parameters,
%   N - SxQ matrix of net input (column) vectors.
%   FP - Struct of function parameters (ignored).
%   and returns A, the SxQ matrix of N's elements squashed into [0, 1].
%
%   LOGSIG('dn',N,A,FP) returns SxQ derivative of A w-respect to N.
%   If A or FP are not supplied or are set to [], FP reverts to
%   the default parameters, and A is calculated from N.
%   LOGSIG('name') returns the name of this function.
%   LOGSIG('output',FP) returns the [min max] output range.
%   LOGSIG('active',FP) returns the [min max] active input range.
%   LOGSIG('fullderiv') returns 1 or 0, whether DA_DN is SxSxQ or SxQ.
%   LOGSIG('fpnames') returns the names of the function parameters.
%   LOGSIG('fpdefaults') returns the default function parameters.
% Examples
%
%   Here is code for creating a plot of the LOGSIG transfer function.
%
%       n = -5:0.1:5;
%       a = logsig(n);
%       plot(n,a)
%
%   Here we assign this transfer function to layer i of a network.
%   net.layers{i}.transferFcn = 'logsig';
%
% Algorithm
%
%       logsig(n) = 1 / (1 + exp(-n))
%
% See also SIM, DLOGSIG, TANSIG.

% Mark Beale, 1-31-92
% Revised 12-15-93, MB
% Revised 11-31-97, MB
% Copyright 1992-2007 The MathWorks, Inc.
% $Revision: 1.1.6.6 $ $Date: 2007/11/09 20:53:04 $

```

```
fn = mfilename;
boiler_transfer
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
Name
```

```
function n = name
n = 'PW Sigmoid';
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
Output range
```

```
function r = output_range(fp)
```

```
r = [0 1];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
Active input range
```

```
function r = active_input_range(fp)
```

```
r = [-4 +4];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
Parameter Defaults
```

```
function fp = param_defaults(values)
```

```
fp = struct;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
Parameter Names
```

```
function names = param_names
```

```
names = {};
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
Parameter Check
```

```
function err = param_check(fp)
```

```
err = '';
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
Apply Transfer Function
```

```
function aa = apply_transfer(nn,fp)
```

```
if nargin < 1, error('Note enough arguments. '); end
```

```
% if nargin==2
```

```
% [nr,nc] = size(n);
```

```
% n = n + b*ones(1,nc);
```

```
% end
```

```
[nr,nc] = size(nn);
```

[illegible]

ÖZGEÇMİŞ

Ahmet Turan ÖZDEMİR 1978’de Malatya’da doğdu. İlk ve orta öğrenimini Malatya’da tamamladı. 1995 yılında Yunus Emre Endüstri Meslek ve Teknik Lisesi, Elektronik Bölümü’nden mezun oldu. 1997 yılında Erciyes Üniversitesi Mühendislik Fakültesi Elektronik Mühendisliği Bölümü’ne kayıt yaptırdı ve sırası ile 1998 yılında İngilizce hazırlık sınıfından, 2002 yılında ise Elektronik Mühendisliği Bölümü’nden mezun oldu. Aynı yıl Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik Anabilim Dalı’nda Yüksek Lisans öğrenimine başladı. 2002-2004 yılları arasında Özkar Enerji Ar&Ge bölümünde donanım tasarım mühendisi olarak görev aldı ve 2004 yılında yüksek lisans öğrenimini tamamladı. Aynı yıl Erciyes Üniversitesi Fen Bilimleri Enstitüsü Elektronik Ana Bilim Dalı’nda doktora programına kayıt yaptırdı ve eş zamanlı olarak Elektrik Elektronik Mühendisliği bölümünde araştırma görevlisi olarak çalışmaya başladı. 2007-2008 yılları arasında Massachusetts Institute of Technology (MIT) Üniversitesi’nde bütünleştirilmiş doktora programı çerçevesinde, MIT Sea Grant College Program bursu ile Autonomous Underwater Vehicle (AUV) Laboratuvarı’nda r-Modem projesinde, donanım tasarım mühendisi olarak görev aldı. Halen Erciyes Üniversitesi’nde görevine devam etmektedir.

İletişim

Adres: Erciyes Üniversitesi Mühendislik Fakültesi
Elektrik-Elektronik Mühendisliği Bölümü
38039 Melikgazi/KAYSERİ

Tel: (0 352) 437 49 01 – 32 233

E-mail 1: aturan@erciyes.edu.tr

E-mail 2: aturanoz@hotmail.com

Kişisel Sayfası: www.aturan.com