



T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**SEZGİSEL ALGORİTMA ÖĞRENMELEİ
YAPAY SİNİR AĞLARI İLE EPİLEPSİ
HASTALIĞININ TEŞHİSİ**

Nesibe YALÇIN

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliğı Anabilim Dalı

Ocak-2012
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Nesibe Yalçın tarafından hazırlanan Sezgisel algoritma Öğrenmeli Yapay Sinir Ağları ile Epilepsi Hastalığının Teşhisi adlı tez çalışması 10/01/2012 tarihinde aşağıdaki jüri tarafından oy birliği / ~~oy çokluğu~~ ile Selçuk Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS / ~~DOKTORA~~ TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Başkan

Yrd. Doç. Dr. Ömer Kaan BAYKAN

Danışman

Yrd. Doç. Dr. Gülay TEZEL

Üye

Yrd. Doç. Dr. Humar KAHRAMANLI

İmza

.....
Ö. Kaan

.....
Gülay

.....
Humar

Yukarıdaki sonucu onaylarım.

Prof. Dr. Bayram Sade
FBE Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Nesibe YALÇIN

Tarih: 11.01.2012

ÖZET

YÜKSEK LİSANS TEZİ

SEZGİSEL ALGORİTMA ÖĞRENMELİ YAPAY SİNİR AĞLARI İLE EPILEPSİ HASTALIĞININ TEŞHİSİ

Nesibe YALÇIN

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Yrd.Doç.Dr. Gülay TEZEL
Yrd.Doç.Dr. Cihan KARAKUZU

2012, 70 Sayfa

Jüri

Yrd.Doç.Dr. Gülay TEZEL
Yrd.Doç.Dr. Ömer Kaan BAYKAN
Yrd.Doç.Dr. Humar KAHRAMANLI

Bu tez çalışmasında, Elektroansefalogram (EEG) kayıtlarının değerlendirilip epilepsi teşhisinin yapılabilmesi amaçlanmıştır. Bunun için sınıflandırma aracı olarak son yıllarda sıkça kullanılan ve etkin bir sınıflandırma tekniği olarak bilinen yapay sinir ağları kullanılmıştır. Öğrenme algoritması olarak 1995 yılında ortaya atılan bir sezgisel algoritma olan parçacık sürüsü optimizasyonu (PSO) yöntemi tercih edilmiştir. Yapay sinir ağını eğitmek için kullanılan bu yöntem, geleneksel yöntemlerden biri olan geri yayılım algoritması ile kıyaslanmış ve sonuçlar yorumlanmıştır. PSO algoritması kullanılması durumunda, sınıflandırma doğruluğu %98 bulunmuştur. PSO tabanlı sinir ağı modeli, geri yayılım sinir ağı (BPNN) modeline göre daha iyi sınıflandırma gerçekleştirmiştir.

7 tane PSO tabanlı yapay sinir ağı (PSONN) modeli (PSONN1, PSONN2, PSONN3, PSONN4, PSONN5, PSONN6 ve PSONN7) tanımlanmıştır. BPNN ve PSONN modellerinin algoritmasının yazılımı Matlab programlama dili ile gerçekleştirilmiştir. Modeller içerisinde sınıflandırma için en uygun olan modeller, PSONN3 ve PSONN7 olarak belirlenmiştir. Ayrıca, farklı hücre, iterasyon ve parçacık sayıları denenmiş, en uygun değerler sırasıyla 3, 200 ve 30 olarak bulunmuştur.

Anahtar Kelimeler: Eylemsizlik ağırlığı, geri yayılım algoritması, parçacık sürüsü optimizasyonu, sezgisel algoritmalar, sınırlama katsayısı, yapay sinir ağları.

ABSTRACT

MS THESIS

HEURISTIC ALGORITHM BASIS ARTIFICIAL NEURAL NETWORKS FOR EPILEPSY DETECTION

Nesibe YALÇIN

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF
SELÇUK UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE / DOCTOR OF PHILOSOPHY
IN COMPUTER ENGINEERING**

**Advisor: Asst.Prof.Dr. Gülay TEZEL
Asst.Prof.Dr. Cihan KARAKUZU**

2012, 70 Pages

Jury

**Asst.Prof.Dr. Gülay TEZEL
Asst.Prof.Dr. Ömer Kaan BAYKAN
Asst.Prof.Dr. Humar KAHRAMANLI**

In this thesis, it was aimed that making epilepsy diagnosis by evaluation of Electroencephalogram (EEG) records. For this, as the classifier, a technique, which is frequently used and known as an active classification technique, artificial neural networks is used. As training algorithm, particle swarm optimization (PSO) method, which is a proposed heuristic algorithm in 1995, is preferred. This method which is proposed to evolve the artificial neural network is compared with back propagation algorithm, which is one of the traditional methods and the results are interpreted. In case of using the PSO algorithm, the classification accuracy was %98. PSO based neural network model (PSONN) has a better classification than back-propagation neural network (BPNN) model.

Seven PSO based neural network models is described. The designed algorithm of PSONN's and BPNN models was developed via MATLAB programming language. In these models, PSONN3 and PSONN7 is determined as appropriate models for the classification. In addition, different number of neuron, iteration and particle have been tried and most appropriate values of these are 3, 200 and 30, respectively.

Keywords: Artificial neural networks, back propagation algorithm, constriction factor, heuristic algorithms, inertia weight, particle swarm optimization.

ÖNSÖZ

Tez çalışmamın hazırlanması boyunca bilgilerini ve desteklerini esirgemeyen, bilimsel disiplin ve bakış açısı kazanmamı sağlayan değerli danışmanlarım Yrd.Doç.Dr. Gülay TEZEL ve Yrd.Doç.Dr. Cihan KARAKUZU' ya, yüksek lisans eğitimim süresince bana emeği geçen bütün Selçuk Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Ana Bilim Dalı Öğretim Üyeleri' ne ve tezimi bitirmem için beni yüreklendiren dostlarıma ve çalışma arkadaşlarıma teşekkürü bir borç bilirim.

Maddi imkansızlıklara rağmen eğitim görmemi sağlayan, sevgilerini ve güvenlerini hiçbir zaman esirgemeyen babam ve annem başta olmak üzere canım aileme saygılarımı ve teşekkürlerimi sunarım.

Nesibe YALÇIN
KONYA-2012

İÇİNDEKİLER

ÖZET	iv
ABSTRACT	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR	ix
1. GİRİŞ	1
1.1. Çalışmanın Amacı ve Önemi	3
1.2. Kaynak Araştırması	3
2. MATERYAL VE YÖNTEM	8
2.1. EEG Veri Seti.....	8
2.2. Yapay Sinir Ağları.....	8
2.2.1. Yapay Sinir Ağlarının Genel Özellikleri.....	10
2.2.2. Yapay Sinir Ağlarının Yapısı	11
2.2.3. Yapay Sinir Ağlarının Elemanları	12
2.2.3.1. Girdiler	12
2.2.3.2. Ağırlıklar	13
2.2.3.3. Toplama Fonksiyonu.....	13
2.2.3.4. Aktivasyon Fonksiyonu.....	13
2.2.3.5. Çıktılar	15
2.2.4. Yapay Sinir Ağlarının Sınıflandırılması	15
2.2.4.1. Yapılarına Göre Yapay Sinir Ağları	16
2.2.4.2. Öğrenme Algoritmalarına Göre Yapay Sinir Ağları.....	16
2.3. Sezgisel Algoritmalar	17
2.3.1. Parçacık Sürüsü Optimizasyonu	17
2.3.1.1. Parçacık Sürüsü Optimizasyonu Algoritması.....	18
2.3.1.2. Parçacık Sürüsü Optimizasyonu Parametre Kontrolü.....	20
3. YAPAY SİNİR AĞLARININ EĞİTİMİ	24
3.1. Geri Yayılım Algoritması ile YSA' nın Eğitimi	24
3.2. PSO ile YSA' nın Eğitimi.....	27
4. UYGULAMA	30
4.1. Geri Yayılım Algoritması ile YSA' nın Gerçeklenmesi.....	31
4.2. Çeşitli PSO Yapıları ile YSA' nın Gerçeklenmesi.....	34
5. SONUÇLAR VE ÖNERİLER	64
5.1. Sonuçlar	64

5.2. Öneriler	65
KAYNAKLAR	66
ÖZGEÇMİŞ.....	70

SİMGELER VE KISALTMALAR

Simgeler

α	: Sabit bir sayı
c_1, c_2	: Öğrenme faktörleri
d	: Parçacık boyutu
g_{best_i}	: Sürünün en iyi konumu
i	: Parçacık indisi
j	: Parçacık boyut indisi
k	: İterasyon sırası
n	: İterasyon sayısı
N	: Parçacık sayısı
$p_{best_{ij}}$: Parçacığın yerel en iyi konumu
r_1, r_2	: Normal dağılımlı rastgele sayı
t	: Azaltma katsayısı
x_{ij}	: Parçacık konumu
v_{ij}	: Parçacık hızı
V_{max}	: Maksimum parçacık hızı
w	: Eylemsizlik ağırlığı
w_{max}	: Maksimum eylemsizlik ağırlığı
w_{min}	: Minimum eylemsizlik ağırlığı
χ	: Sınırlama katsayısı
λ	: Öğrenme katsayısı
δ_c	: c. hücrenin eğimi

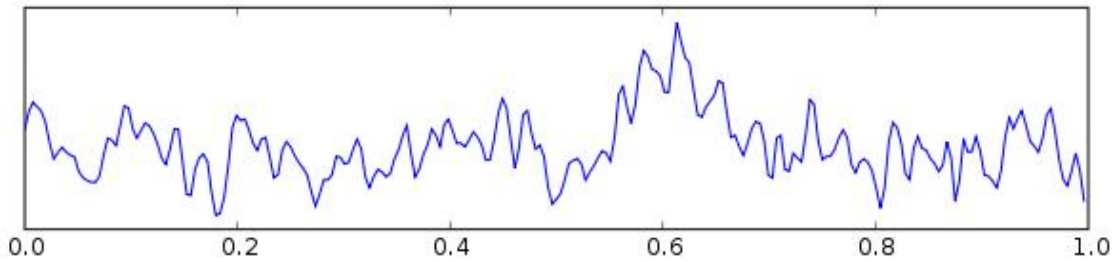
Kısaltmalar

AAF	: Adaptif Aktivasyon Fonksiyonu (Adaptive Activation Function)
AR	: Özbağlanımlı (Auto - Regressive)
ART	: Adaptif Rezonans Teori
BP	: Geri Yayılım (Back-Propagation)
BPNN	: Geri Yayılımlı Sinir Ağı (Back-Propagation Neural Network)
EEG	: Elektroansefalogram
EKG	: Elektrokardiyogram
EKK-DVM	: En Küçük Kareler Destek Vektör Makinesi
EMG	: Elektromiyogram
HFD	: Hızlı Fourier Dönüşümü
LVQ	: Doğrusal Vektör Parçalama (Linear Vector Quantization)
MIT-BIH	: Massachusetts Institute of Technology - Beth Israel Hospital
MLP	: Çok Katmanlı Algılayıcı (Multi Layer Perceptron)
MSE	: Karese Ortalama Hata (Mean Square Error)
PSO	: Parçacık Sürüsü Optimizasyonu (Particle Swarm Optimization)
PSOINN	: PSO Tabanlı Sinir Ağı
RTFYSA	: Radyal Tabanlı Fonksiyon Yapay Sinir Ağı
RVM	: Bağlantı Vektör Makinesi (Relevance Vector Machine)
YSA	: Yapay Sinir Ağları (Artificial Neural Networks)

1. GİRİŞ

Richard Caton, 1875 yılında hayvanlar üzerinde yaptığı deneyler sonucunda beyinde bir takım elektriksel faaliyetlerin varlığını tespit etmiştir. Böylece Elektroansefalogram (EEG)' in tarihsel gelişimi başlamıştır. Hans Berger, 1929 yılında insan beynindeki elektriksel aktivitenin varlığını ilk defa ortaya koymuş ve 1930 yılında EEG adı verilen bu dalgaların gözün açılıp kapanmasıyla değiştiğini göstermiştir. EEG sinyallerinin varlığı, elektronikte meydana gelen gelişmeler sonucunda bilimsel şekilde ortaya çıkarılabilmektedir (Toprak, 2007).

EEG işaretleri periyodik değildir; genlik, faz ve frekansları sürekli değişmektedir. EEG işaretlerinin Şekil 1.1.' de görüldüğü gibi değişim şekli çok karmaşıktır ve yorumlanması zordur. Kafa üzerinden algılanan EEG' lerin genliği tepeden tepeye 1–100 μV ve frekans bandı 0,5–100 Hz' dir (Toprak, 2007; Yazgan ve Korürek, 1996).



Şekil 1.1. Bir EEG sinyali örneği

Halk dilinde “epilepsi”, “sara” hastalığı olarak bilinmektedir ve beyinde meydana gelen önemli rahatsızlıklardan biridir. Özellikle epileptik deşarjların meydana gelmesi esnasında kaydedilen EEG' nin içerdiği dalga şekilleri diğer bazı beyin rahatsızlıklarındaki dalga şekilleriyle benzerlik gösterebilmektedir. Bu sebeple, epilepsi hastalığı kolay tespit edilememektedir. Bu yüzden teşhisi yapacak doktorun iyi bir gözlemci ve deneyimli olması gerekmektedir (Barışçı ve Müldür, 2003).

Son zamanlarda, bu zorlukları ortadan kaldırmak için yapay zeka yöntemleri kullanılarak EEG işaretlerinin tanınması yönünde oldukça fazla çalışma yapılmaktadır (Avşar ve ark., 2009; Barışçı ve Müldür, 2003; Güler ve Übeyli, 2004; Han ve Sun, 2010; Harikumar ve Narayanan, 2003; Özbeyaz, 2010). Bir yapay zeka yöntemi olan Yapay Sinir Ağları (YSA) da EEG işaretlerinin sınıflandırılmasında yaygın olarak

kullanılmaktadır (Akın ve ark., 2001; Batar, 2005; Bezobrazova ve Golovko, 2007; Delice, 2008; Hema ve ark., 2008; Sezer, 2008; Sivasankari ve Thanuskodi, 2009; Subaşı ve ark., 2004; Şahin ve ark., 2008; Tezel, 2007; Tezel ve Özbay, 2007; Toprak, 2007).

YSA ile EEG sinyallerinin büyük miktardaki örnekleri kısa süre içerisinde analiz edilerek bir anormallik olup olmadığı bulunabilir. YSA, biyolojik sinir hücresinden esinlenerek gerçekleştirilmiş bir yapıdır. Örnekler yardımı ile insan beyninin özelliklerinden biri olan öğrenme işlemini gerçekleştirir. Örneklerden elde ettikleri bilgiler ile deneyimlerini oluşturur ve daha sonra benzer konularda benzer kararları verirler. Matematiksel olarak bağlantısı kurulamayan, kurulsu dahi çözümlenmesi çok karmaşık olan problemlerin çözümlenmesine olanak sağlarlar. YSA, EEG sinyallerinin çok sayıdaki örneklerinin her saniyesini analiz edebilecek kadar hızlı yapıya sahiptir (Sezer, 2008).

Son zamanlarda, yapay zeka yöntemlerinin yanında, sezgisel optimizasyon algoritmaları da hem yapay zeka yöntemlerinin başarısını ve/veya hızını artırmak için hem de tek başlarına optimizasyon yapabilmek için kullanılmaktadırlar (Bağış ve Çetin, 2009; Der ve ark., 2008; Doğan, 2009; Gallad ve ark., 2002; Gözde ve ark., 2010; Kulluk, 2009; Liu ve ark., 2008; Özbeyaz, 2010; Parsopoulos ve Vrahatis, 2002; Tewolde ve Hanna, 2007; Wang ve ark., 2007). Sezgisel yöntemlerden biri olan Parçacık Sürüsü Optimizasyonu (PSO), gerçekleştirilmesinin kolaylığı, ayarlanması gereken parametre sayısının azlığı ve reel sayılarla çalışıyor olması gibi nedenlerle pek çok alanda başarı ile uygulanmıştır. Bunlardan birisi de yapay sinir ağlarının eğitimidir (Calvalho ve Ludermer, 2007; Delice, 2008; Gu ve ark., 2009; Hema ve ark., 2008; Ninomiya ve Zhang, 2008; Tamer ve Karakuzu, 2006).

Bu çalışmada, epilepsi teşhisini hızlı ve doğru bir şekilde gerçekleştirmek için hastalardan alınan EEG sinyalleri normalizasyon işlemine tabii tutulmuş ve YSA ile sınıflandırılma işlemi gerçekleştirilmiştir. YSA' nın eğitilmesinde farklı PSO versiyonları kullanılmış, performansı geri yayılım algoritması kullanan YSA ile karşılaştırılmıştır.

Bu çalışmanın birinci bölümünde, genel bir giriş yapılarak tez çalışmasının amacı ve önemi verilmiş, literatürdeki yeri incelenmiştir. İkinci bölümde, bu çalışmada kullanılan materyal ve metotlara değinilmiştir. YSA ve PSO genel bir bakış açısıyla verilmiştir. Üçüncü bölümde, YSA' nın geri yayılım algoritması ve değişik PSO yöntemleri ile eğitilmesi üzerinde durulmuş, algoritmaları açıklanmış, akış şemaları

verilmiştir. Dördüncü bölümde ise gerçekleştirilen uygulamalar hakkında bilgi verilmiş, elde edilen sonuçlar tablolar ile sunulmuş ve karşılaştırılma işlemleri yapılmıştır. Son bölümde ise tez çalışması özetlenerek sonuç ve değerlendirme yapılmıştır.

1.1. Çalışmanın Amacı ve Önemi

Tıbbi karar verme, veri sınıflama işlemi olarak incelenebilmektedir. Çok yönlü karmaşık bir işlemdir ve doğru teşhis yapılması önemlidir. Bunun için uygun verileri bulmak, verilerin özelliklerini çıkarmak ve yeni veriyi analiz etmek gerekmektedir. Verinin karmaşıklığı ve boyutunun artmasına bağlı olarak bilgisayar kullanımına ihtiyaç duyulmaktadır. Bilgisayar destekli veri sınıflama, YSA uygulamaları arasında yer almaktadır (Güler ve Übeyli, 2004).

Epilepsi teşhisi önemli bir konudur, hastanın ve EEG' sinin gözlenmesinin yanı sıra bazı klinik bilgilere de gereksinim duyulmaktadır. Hastaları epileptik veya hasta değil şeklinde sınıflandıran bir yapay sinir ağı, doktor için çok yararlı bir karar verme aracı olacaktır (Subaşı ve ark., 2004).

Bu çalışmada, EEG sinyallerinin Sezgisel Algoritma tabanlı Yapay Sinir Ağları ile otomatik olarak değerlendirilip epilepsi teşhisinin yapılabilmesi amaçlanmıştır. Kendisine gösterilen örnekleri genellemesi, daha sonra hiç görmediği örnekler verildiğinde öğrendiği bilgileri kullanarak karar verebilmesi gibi özellikleri (Batar, 2005) sebebiyle YSA tez çalışmasında kullanılmak üzere tercih edilmiştir.

İlk olarak, EEG verileri üzerinde istatistiksel işlemler yapılarak öznelik çıkarımı yapılmış ve böylece verilerin boyutları azaltılmıştır. Daha sonra elde edilen veriler, öğrenme algoritması olarak çeşitli PSO versiyonlarını kullanan YSA ve geri yayılım algoritmasını kullanan YSA' nın girişlerine uygulanmıştır. Elde edilen ağların çalışması karşılaştırılmış ve sınıflandırma performansları incelenmiştir. Değişik YSA yapıları denenmiş, en hızlı ve en başarılı olan YSA yapısının bulunması hedeflenmiştir. Uygulamanın gerçekleştirilmesinde MATLAB® program paketi kullanılmıştır.

1.2. Kaynak Araştırması

PSO algoritmasının pek çok optimizasyon problemine başarı ile uygulanması (Delice, 2008) ile birlikte literatürde yapay sinir ağlarının eğitiminde kullanılmasına ilişkin uygulama sayıları da artmaya başlamıştır. Bunlardan bazıları;

Akın ve Ark. (2001), EEG sinyallerini sınıflandırmak için YSA ve dalgacık (wavelet) dönüşümü kullanarak yeni bir yaklaşım geliştirmişlerdir. EEG sinyallerini dalgacık dönüşümü ile delta, teta, alfa ve beta bileşenlerine ayırdıktan sonra sinir ağının girişlerine vermişlerdir. Ağ çıkışlarının doğruluğu (epileptik için %97, sağlıklı için %98, patolojik için %93) çok yüksek bulunmuştur.

Barışçı ve Müldür (2003), doğru ve hızlı bir teşhis gerçekleştirebilmek amacıyla epileptik ve hasta olmayan kişilerden alınan her bir EEG sinyaline, Hızlı Fourier Dönüşümü (HFD) spektral analizi uygulamışlar, HFD sonuçlarını sinirsel – bulanık sistem kullanarak sınıflandırmışlardır. Sınıflandırma sonucunda sinirsel – bulanık sistemin teşhise yönelik iyi sonuçlar verdiğini gözlemlemişlerdir.

Batar (2005) çalışmasında, EEG işaretlerini uyanıklık, uyuklama ve uyku hallerinde gösterdikleri farklılıkları değerlendirip YSA kullanarak bilgisayar destekli bir otomatik sınıflandırma yapmayı amaçlamış ve bunu başarı ile sağlamıştır.

Bezobrazova ve Golovko (2007) yaptıkları çalışmada, epilepsi tespiti için sinir ağı tekniğini göstermişlerdir. Lyapunov metodunun hesaplanması için hangi YSA' nın daha iyi olduğunu tespit etmeye çalışmışlardır. Bunun için, çok katmanlı algılayıcı (MLP), Elman yinelemeli YSA ve Radyal Tabanlı Fonksiyon Yapay Sinir Ağı (RTFYSA) incelenmiştir. Bu ağların başarımı konusunda belirgin bir fark olmadığı rapor edilmiştir.

Calvalho ve Ludermir (2007)' in çalışmalarında, MLP mimarileri ve bağlantı ağırlıklarının küresel optimizasyonu için parçacık sürüsü optimizasyon algoritmasına başvurulmuştur. Önerilen algoritmayı değerlendirilebilmesi için, tıp alanında (kanser, diyabet, kalp) kriter (benchmark) sınıflandırma problemleri kullanılmıştır. Parçacık Sürüsü Optimizasyonu dayalı bu yaklaşımın, ağırlıklar ve MLP yapay sinir ağları mimarilerinin optimize edilebilmesi için geçerli bir alternatif olduğu gösterilmiştir.

Delice (2008)' nin PSO ile YSA' dan sınıflandırma kuralı çıkarımı konulu tez çalışmasında, eğitilmiş YSA' dan sınıflandırma kuralları çıkarmak için ikili PSO (iPSO) algoritması geliştirmiştir. Daha sonra test veri kümeleri üzerinde yaptığı analizlerle, geliştirilen algoritmanın doğru ve etkin sınıflandırma kuralları üretebildiğini göstermiştir.

Doğan (2009) tez çalışmasında, EKG vurularındaki anormalliklerin tespit edilmesi amacı ile öğrenme algoritması olarak PSO' yu kullanan RTFYSA önermiştir. Ağ eğitimi için, MIT-BIH aritmi veritabanından alınan elektrokardiyogram (EKG) kayıtları üzerinde düzeltmeler yapılmış ve PSO algoritması kullanılarak, RTFYSA

eđitilmiş ve bulunan optimum ađ yapısı sunulmuştur. K-ortalamlar, Kohonen ve k-en yakın komşu yöntemleri ile kıyaslama yapılmış ve önerilen yöntemin ađın sınıflama başarısından ödün vermeden diđer yöntemlere nazaran çok daha hızlı sınıflama sonuçları verdiđi göstermiştir.

Güler ve Übeyli (2004), dalgacık dönüşümü ile EEG işaretlerinden çıkarılan öznitelik vektörleri üzerinde istatistiksel işlemlerin gerçekleştirilmesi ile ilgili bir çalışma yapmışlardır. Bu çalışmada, dalgacık sinir ađıyla yapılan EEG sınıflandırmasının daha iyi sonuçlar verdiđi ve bu sonuçların hastalık teşhisinde kullanılabileceđi görülmüştür.

Güler ve Übeyli (2006)' nin diyabet hastalığının teşhisi için yaptıkları çalışmada, dört farklı algoritma (geri yayılım, delta-bar-delta, genişletilmiş delta-bar-delta ve hızlı yayılım) ile eğitilen çok katmanlı algılayıcı sinir ađları kullanılmış ve en başarılı algoritma belirlenmeye çalışılmıştır. Hızlı yayılım algoritmasının en iyi başarımı verdiđi gösterilmiştir.

Han ve Sun (2010) yaptıkları çalışmada, bağlantı vektör makinesi (RVM) ve öz bağlantımlı (AR) modeline dayalı yeni bir EEG sinyal sınıflandırma metodu amaçlamışlar ve üç aşamada (AR modeline dayalı EEG sinyallerinden özellik çıkarma, özellik seçimi, sınıflandırma) gerçekleştirmişlerdir.

Hema ve ark. (2008) yaptıkları çalışmada, parçacık sürüsü optimizasyonu sinir ađı kullanarak EEG işaretlerinin sınıflandırılması için bir sınıflandırma algoritması önermişlerdir. Beş zihinsel iş zamanında kayıt edilen EEG sinyallerinden özellikler çıkartılmış ve bu özellikler ikili kombine edilerek sinir ađlarının eğitilmesi ve test edilmesi için kullanılmıştır. 5 ve 10 gizli katman hücre sayıları kullanılmış, 5 hücre sayısına sahip sinir ađında daha iyi sınıflandırma performansı verdiđi gözlenmiştir. Ayrıca elde edilen sonuçlar PSO' nun performansını onaylamıştır.

Kennedy ve Eberhart (1995), kuş sürülerinin sosyal davranışını model alan sezgisel tabanlı bir optimizasyon yöntemi olan parçacık sürüsü optimizasyonu algoritmasını geliştirmişlerdir.

Ninomiya ve Zhang (2008) geri beslemeli sinir ađlarının eğitimi için geliştirdikleri teknikte, PSO' nun küresel optimizasyon yeteneđi ile genelleştirilmiş Newton yönteminin hızlı yakınsama yeteneđi kullanılmıştır. Başlangıç değerler ne olursa olsun geleneksel küresel optimizasyon yöntemlerinden daha yüksek küresel yakınsama sağlandığı belirtilmiştir.

Özbeyaz çalışmasında (2010), uzman hekimlere teşhislerde yardımcı olacak otomatik bir sistemin tasarlanmasını amaçlamıştır. En Küçük Kareler Destek Vektör Makinesi (EKK-DVM) ile Elektromiyogram (EMG) ve EEG işaretlerini sınıflandırmış, EKK-DVM için parametre seçiminde PSO algoritmasını kullanmıştır.

Sezer (2008) tez çalışmasında, epilepsi teşhisi için EEG işaretlerinin dalgacık analizi ile incelendikten sonra çeşitli YSA ile sınıflandırılmasını amaçlamıştır. EEG işaretleri dalgacık dönüşümü ile frekans alt-bantlarına ayrılmış ve bu alt-bantlardan öznitelik vektörleri elde edilmiş, elde edilen öznitelik vektörlerinin sayısı azaltıldıktan sonra MLP, Elman, LVQ (Doğrusal Vektör Parçalama) gibi değişik YSA' nın girişlerine verilmiştir. MLP ağ yapısı dışındaki YSA' larla hızlı öğrenme gerçekleştirildiği görülmüş ve MLP yapılarından iki katmanlı olanların tek katmanlı olanlara göre daha başarılı oldukları belirlenmiştir.

Sivasankari ve Thanushkodi (2009), EEG sinyallerinden otomatik olarak teşhis yapmak için etkili bir yaklaşım amaçlanmıştır. İlk olarak bağımsız bileşen analizi ile teşhis için gerekli bileşenler elde edilmiştir. Daha sonra bu bileşenler ile YSA eğitilmiştir. Elde edilen sonuçlar, önerilen yaklaşımın EEG sinyallerinde epileptik krizin varlığını algılamada etkin olduğunu göstermiştir.

Subaşı ve arkadaşlarının (2004) yaptıkları çalışmada, EEG kayıtlarında epilepsi varlığını belirlemede sınıflandırıcı araç yeni bir dalgacık sinir ağı geliştirilmiştir. Bu metotta AR spektrumlar dalgacık sinir ağına giriş olarak verilmiş, çıkışta epilepsi var veya yok şeklinde çıkış alınmıştır. Geri yayımlı öğrenme algoritması ile denenmiş ve sınıflandırma doğruluğunun arttığı gözlemlenmiştir.

Şahin ve Ark. (2008)' nın yaptıkları çalışma, YSA kullanılarak EEG sinyallerine göre kısmi epilepsi gruplarını sınıflandıran ilk çalışmadır. Bu çalışmada, sadece EEG sinyallerinden değil hastaların diğer demografik özelliklerini kullanarak MLP sinir ağını eğitmişlerdir.

Tamer ve Karakuzu (2006)' nun çalışmalarında, EXOR probleminin çözülebilmesi için gerekli ağ parametreleri PSO algoritması ile geri yayılım algoritmasında hızlı bir şekilde belirlenmiştir. Yapay sinir ağlarının eğitimi için kullanılan geri yayılım öğrenme algoritması yerine PSO' nun kullanılabileceği ifade edilmiştir.

Tewolde ve Hanna (2007) yaptıkları çalışmada, Wisconsin göğüs kanseri verilerini sınıflandırma amacıyla kullanmışlardır. Verilerin sınıflandırılmasında tek ve çoklu yüzey tabanlı veri ayırma metotlarının uygulanması için PSO' nun etkinliğinden

faydalanmışlardır. Verileri eğitim ve test kısmında kullanılmak üzere ikiye ayırdıktan sonra, PSO tabanlı sınıflandırıcı ile eğitim ve test işlemlerini gerçekleştirmişlerdir. Daha sonra tek ve çoklu yüzeylerde performanslarını karşılaştırmışlardır.

Tezel (2007) tez çalışmasında, EEG ve EKG verilerinin sınıflandırılması amacıyla üç farklı Adaptif Aktivasyon Fonksiyonu (AAF) YSA modeli ve bir adet çok katmanlı YSA modeli geliştirmiştir. Dalgacık dönüşümü ve temel bileşen analizi yöntemlerini kullanarak verilerden özellik çıkartmış, sınıflandırıcının girişlerine uygulamış ve modellerin sınıflandırma performansını karşılaştırmıştır. Geliştirilen AAFYSA modellerinin özellikle eğitime zamanı, iterasyon sayısı ve ağ mimarisinin karmaşıklığı konusunda literatürdeki diğer çalışmalardan daha başarılı oldukları göstermiştir.

Toprak (2007), yüksek lisans tez çalışmasında, EEG kayıtlarının otomatik olarak değerlendirilip epilepsi teşhisinin yapılabilmesini amaçlamıştır. Teşhis işlemi, öznitelik çıkarma/seçme ve sınıflama olmak üzere ikiye ayırmıştır. Öznitelik vektörlerinin belirlenmesinde ayrık dalgacık dönüşümünü, sınıflama için YSA' yı kullanılmıştır. Farklı dalgacıkların, değişik öğrenme algoritmalarıyla eğitilen MLP sinir ağları ve RTFYSA üzerindeki etkisini inceleyerek yüksek sınıflama doğruluğu elde etmiştir.

Wang ve ark. (2007) çalışmalarında, veritabanında etkili sınıflandırma kuralı çıkarmak için PSO tabanlı bir çıkarım algoritması amaçlamışlardır. Daha az kural ile daha yüksek doğruluk hedeflemişler ve 6 veri kümesi üzerinde eğitim ve test işlemlerini gerçekleştirmişlerdir. PSO' dan başka genetik algoritma ve karınca koloni algoritması kullanılarak performansları karşılaştırılmıştır. Sonuç olarak PSO' nun kullanılması durumunda daha yüksek doğruluk elde edildiği, daha kısa sürede daha az sayıda basit kural listesi verdiği görülmüştür.

Gu ve arkadaşlarının (2009) yaptıkları çalışmada, geri beslemeli sinir ağlarının eğitilmesinde öğrenme algoritması olarak PSO çeşitleri kullanılmış ve algoritması önerilmiştir. Önerilen algoritmanın sürü çeşitliliğini iyileştirdiği ve yakınsama performansını artırdığı gözlenmiştir.

2. MATERYAL VE YÖNTEM

Bu çalışmada, epileptik ve hasta olmayan kişilerden alınan EEG verilerinin sınıflandırılması için YSA ve sezgisel bir optimizasyon yöntemi kullanılmıştır.

2.1. EEG Veri Seti

EEG, beyindeki hastalıkların teşhisinde bir yardımcı olarak kullanılır. Epilepsinin teşhisinde çok kullanışlıdır ve epileptik nöbetlerin sınıflandırılmasına izin verir. Bu çalışmada, Andrzejak ve ark. (2001) 'da tanımlanan genel kullanıma açık EEG verisi ile uygulamalar yapılmıştır. Bu bölümde, verilerle ilgili olarak kısa bir tanımlama yapılmakla birlikte daha fazla bilgi için Andrzejak ve ark. (2001) referans gösterilebilir.

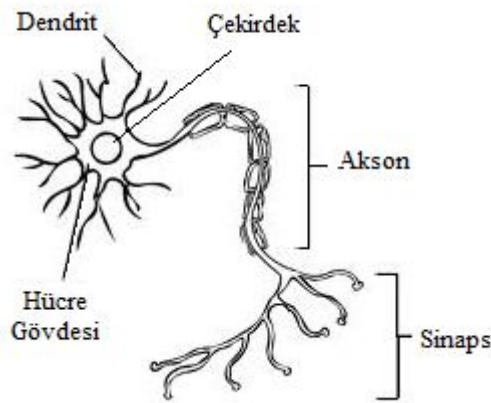
Veri setinin tamamı, her biri 100 tek kanal EEG segmentinden oluşan ve A-E ile ifade edilen 5 setten oluşmaktadır. Bu segmentler, artefaktların görsel olarak tespit edilmesinden sonra, çok kanallı sürekli EEG kayıtlarından seçilmiştir. Set A ve Set B, 5 sağlıklı gönüllüden sırasıyla gözler açık ve kapalı iken yapılmış yüzey elektrotlu EEG kayıtlarından alınan segmentlerden oluşmuştur. D setindeki segmentler, epileptojenik bölgeden ve set C beynin diğer hemisferindeki hipokampal oluşumdan kayıt edilmişlerdir. Set C ve Set D, atak olmayan aralıktan ölçülen aktiviteleri içerirken, set E sadece epileptik atakların bulunduğu aktiviteleri kapsamaktadır. Bütün EEG kayıtları, aynı 128-kanal yükselteç sistemi ile ortak toprak referans kullanarak kaydedilmiş ve 173.61 saniyedeki örnek sayısı ile 12 bit çözünürlükte sayısallaştırılmıştır. Bant geçiren filtre 0.53-40 Hz olarak ayarlanmıştır (Tezel ve Özbay, 2007; Andrzejak ve ark., 2001). Bu çalışmada A ve E olmak üzere iki veri seti kullanılmıştır.

2.2. Yapay Sinir Ağları

Yapay Sinir Ağları (YSA), insan beyninden yola çıkılarak geliştirilmiş, ağırlıklı bağlantılarla birbirine bağlı olan ve her biri kendi belleğine sahip işlem elemanlarından oluşan paralel ve dağıtılmış bilgi işleme yapılarıdır. YSA, biyolojik sinir ağlarını taklit eden bilgisayar programlarıdır (Elmas, 2007).

YSA' nın çalışmasını anlayabilmek için öncelikle biyolojik sinir sisteminin yapısına bakılmalıdır, çünkü YSA, biyolojik sinir ağlarının bir modellemesidir.

Biyolojik sinir ağıları, sinir hücrelerinin bir araya gelmesi sonucu oluşur. Biyolojik bir sinir hücresi (Şekil 2.1.) bir gövde, bir akson, çok sayıda sinir ucu (dendrit) ve akson ile diğer sinir hücresinin sinir ucu arasında kalan ince uzantılar (sinaps) olmak üzere dört bölümden oluşmaktadır. Dendritler, bağlı olduğu diğer sinir hücrelerinden veya duyu organlarından gelen sinyalleri çekirdeğe iletir. Çekirdek dendritten gelen sinyalleri bir araya toplar ve aksona iletir. Toplanan bu sinyaller, akson tarafından işlenerek sinapslara gönderilir. Sinapslar da yeni üretilen sinyalleri diğer sinir hücrelerine iletir (Kakıcı, 2009).



Şekil 2.1. Biyolojik bir sinir hücresinin yapısı

Yapay sinir hücreleri, gerçek sinir hücrelerinin simüle edilmesiyle elde edilir. Yapay sinir hücreleri, aralarında bağ kurarak yapay sinir ağlarını oluştururlar. Yapay sinir hücrelerinin biyolojik sinir hücreleri ile olan benzerliğini ifade eden bölümleri Çizelge 2.1. (Sağiroğlu ve ark., 2003)' de verilmiştir.

Çizelge 2.1. Biyolojik sinir hücresi ile yapay sinir hücresinin karşılaştırılması

Biyolojik Sinir Hücresi	Yapay Sinir Hücresi
Akson	Çıktı
Dendrit	Toplama Fonksiyonu
Çekirdek	Aktivasyon fonksiyonu
Sinaps	Ağırlıklar

2.2.1. Yapay Sinir Ağlarının Genel Özellikleri

- **Öğrenme:** YSA' nın öğrenebilmesi için, girdi değerleri ile bu girdilere karşılık gelen çıktı değerlerinin veya sadece girdi değerlerinin ağa verilmesi gerekir. Girdi – çıktı arasındaki ilişkiyi ya da girdiler arasındaki ilişkilendirmeyi katmanlar arasında bulunan ağırlıklar ifade etmektedir. YSA' da öğrenme, bu ağırlık değerlerinin en uygun hale getirilmesi sürecidir (Sağiroğlu ve ark., 2003). Bunu gerçekleştirebilmek amacıyla çeşitli öğrenme algoritmaları geliştirilmiştir. Bunlara geri yayılım algoritması, delta bar delta algoritması, Levenberg-Marquardt metodu örnek olarak verilebilir.

- **Genelleme Yapma:** Eğitim işleminin ardından YSA' nın ürettiği çıktıların kabul edilebilir düzeyde olup olmadığını test işlemi yaparak öğrenebiliriz. Test işleminde girdiler ağa verilir, elde edilen çıktılar ile gerçekte olması gereken çıktılar karşılaştırılır. Karşılaştırılma sonucunda elde edilen hata değerleri kabul edilebilir sınırlar içerisinde ise ağın performansı iyi kabul edilir. Böylece ağ, daha önce hiç görmediği girdiler verilse bile en uygun kabul edilebilir çıkışı üretecektir (Sağiroğlu ve ark., 2003).

- **Doğrusal Olamama:** Kullanılan aktivasyon fonksiyonuna bağlı olarak sinir hücreleri doğrusal olmadığı için bu elemanların birbirlerine bağlanması ile oluşan ağda doğrusal değildir. Bu özellik, YSA' ya karmaşık ve doğrusal olmayan gerçek çevre problemlerine kabul edilebilir, etkin çözümler sağlama kabiliyeti verir (Kansu, 2006).

- **Adaptasyon:** Yapay sinir ağları ele alınan problemde meydana gelen parametre ya da tanım gibi değişikliklere göre ağırlıklarını ayarlar. Yani, belirli bir problemi çözmek amacıyla eğitilen YSA, problemdeki değişikliklere göre tekrar eğitilebilir, değişimler devamlı ise gerçek zamanda da eğitime devam edebilir (Sağiroğlu ve ark., 2003).

- **Paralel İşlem Yapma:** YSA, çok sayıda hücrenin çeşitli şekillerde bağlanmasından oluştuğundan paralel dağıtılmış bir yapıya sahiptir ve ağın sahip olduğu bilgi, ağdaki bütün bağlantılar üzerine dağılmış durumdadır (Kulluk, 2009).

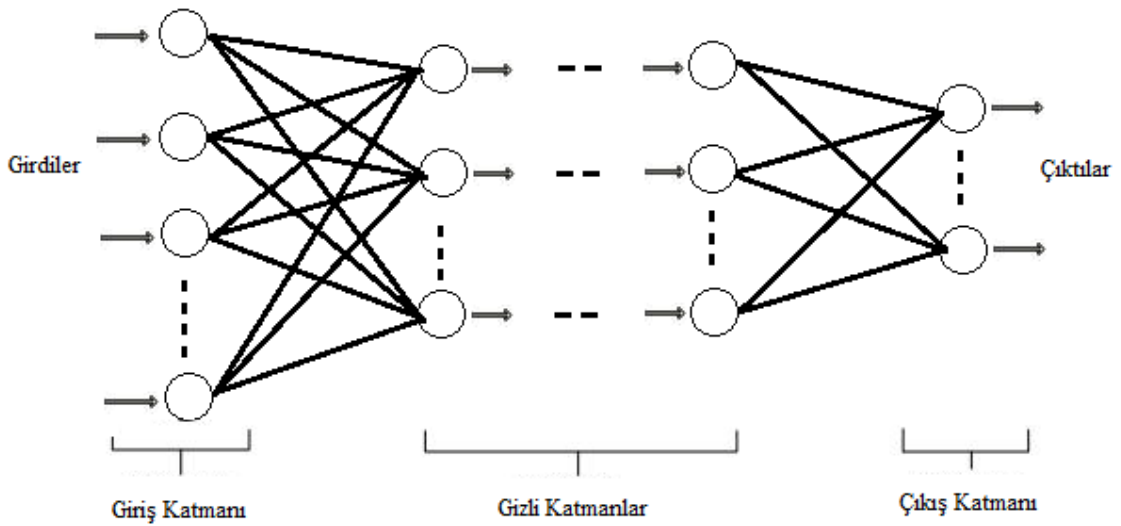
- **Eksik Bilgi ile Çalışabilme:** YSA' da bilgi, ağın tamamında (ağırlıklarda) tutulduğu için girdilerde bulunabilecek herhangi bir gürültü veya eksiklik, bilgi ağırlıklar üzerine dağıtılacağı için tolere edilebilecektir (Sağiroğlu ve ark., 2003).

- **Hata Toleransı:** YSA' nın eksik bilgilerle çalışabilme yetenekleri, hatalara karşı toleranslı olmalarını sağlamaktadır. Ağın bazı hücreleri bozulsa ya da çalışmaz duruma gelse bile ağ çalışmaya devam eder (Öztemel, 2006).

- **Sayısal Veriler ile Çalışabilme:** YSA, sadece sayısal bilgiler ile çalışabilmektedir. Bu yüzden kategorik veriler ile gösterilen bilgilerin sayısal değerler haline getirilmesi gerekmektedir (Öztemel, 2006).

2.2.2. Yapay Sinir Ağlarının Yapısı

YSA, yapay sinir hücrelerinin bir araya gelmesi sonucu oluşan yapılardır. YSA, Şekil 2.2. (Kakıcı, 2009)' te görüldüğü üzere üç bölümde incelenmektedir:



Şekil 2.2. Çok katmanlı bir yapay sinir ağı

- **Giriş Katmanı:** Yapay sinir ağına dış dünyadan girdiler bu katmana gelir. Bu katmanda, girdi sayısı kadar hücre bulunmaktadır ve girdiler herhangi bir işleme uğramadan gizli katmana iletilmektedir. Giriş katmanındaki her bir hücrenin birer adet girdisi ve çıktısı vardır. Ayrıca bu hücreler gizli katmandaki bütün hücrelere bağlıdır.

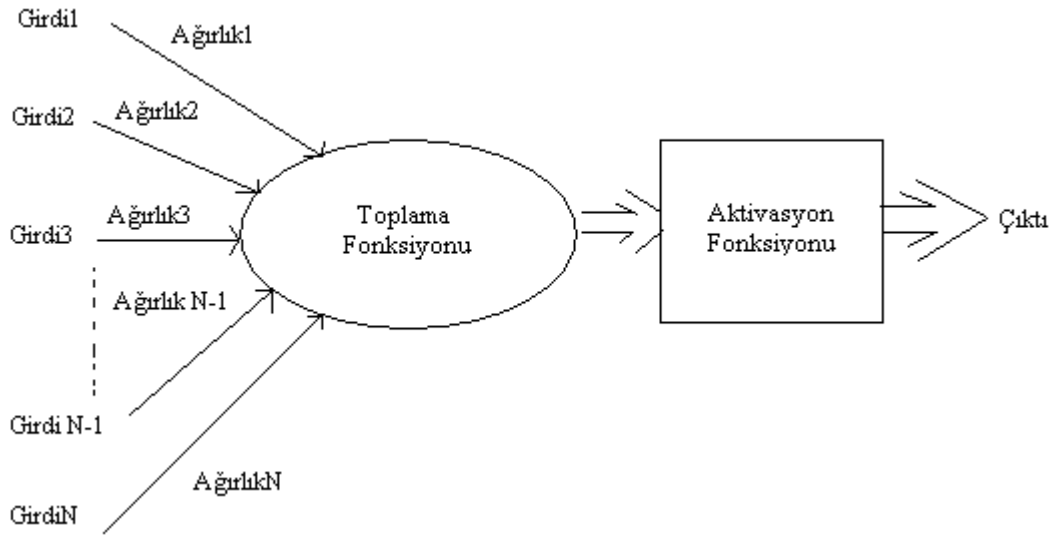
- **Gizli Katmanlar:** Giriş katmanından aldığı bilgiyi işleyerek bir sonraki katmana iletir. Gizli katman sayısı ve gizli katmandaki hücre sayısı ağdan ağa değişebilir. Gizli katmanlardaki hücre sayıları, girdi ve çıktı sayılarından bağımsızdır. Ayrıca birden fazla gizli katman varsa, bu katmanlardaki hücre sayıları aynı olmayabilir. Gizli katman sayısı ve bu katmanlardaki hücre sayısının artması, hesaplama karmaşıklığı ve süresini

artırsa da YSA' nın daha karmaşık problemlerin çözümünde kullanılabilmesini sağlamaktadır.

- **Çıkış Katmanı:** Gizli katmandan gelen bilgiyi işleyer ve giriş katmanına gelen girdiye uygun olarak üretilen çıktıyı dış dünyaya gönderir. Çıkış katmanındaki hücre sayısı birden büyük olabilir. Her bir hücre bir önceki katmandaki bütün hücelere bağlıdır. Her bir çıkış hücresinin bir adet çıktısı vardır (Kakıcı, 2009).

2.2.3. Yapay Sinir Ağlarının Elemanları

YSA elemanları ve bunlar arasındaki ilişkiler, Şekil 2.3. (Öztemel, 2006)' teki gibidir.



Şekil 2.3. Yapay bir sinir hücresinin yapısı

2.2.3.1. Girdiler

Yapay sinir ağlarına dış dünyadan veya diğer bir hücreden gelen bilgilerdir. x_i , girdi değeri, N toplam girdi sayısı ve i, girdi indisi olmak üzere $i=1..N$ ' dir. Giriş değerleri, ağırlıkların belirlenmesinde ve dolayısı ile ağ eğitiminde önemli bir rol oynamaktadır (Altun ve ark., 2002).

2.2.3.2. Ağırlıklar

Hücreler arasındaki bağlantıların sayısal değerini ifade etmektedir. Bir hücreye gelen bilginin değerini ve hücre üzerindeki etkisini gösterir (Öztemel, 2006). w_i , x_i girdisine karşılık gelen ağırlık değeri, N toplam girdi sayısı ve i, indis olmak üzere $i=1..N$ ' dir.

2.2.3.3. Toplama Fonksiyonu

Hücreye gelen girdileri ($X = (x_1 x_2 x_3 \dots x_N)$) ve ağırlıkları ($W = (w_1 w_2 w_3 \dots w_N)$) kullanarak o hücrenin net girdisinin hesaplanmasını sağlar. Çeşitli toplama fonksiyonlarının net girdiyi nasıl hesapladıkları Çizelge 2.2. (Öztemel, 2006)' de verilmiştir.

Çizelge 2.2. Toplama fonksiyonu örnekleri

Net Girdi	Formül	Açıklama
Toplam	$\text{Net Girdi} = \sum_{i=1}^N (w_i x_i)$	Girdiler ağırlık değerleri ile çarpılır ve daha sonra bulunan değerler birbirleri ile toplanarak net girdi hesaplanır.
Çarpım	$\text{Net Girdi} = \prod_{i=1}^N w_i x_i$	Girdiler ağırlık değerleri ile çarpılır ve daha sonra bulunan değerler birbirleri ile çarpılarak net girdi hesaplanır.
Maksimum	$\text{Net Girdi} = \max(w_i x_i), i=1..N$	Girdiler ağırlıklar değerleri ile çarpıldıktan sonra en büyüğü yapay sinir hücresinin net girdisi olarak kabul edilir.
Minimum	$\text{Net Girdi} = \min(w_i x_i), i=1..N$	Girdiler ağırlık değerleri ile çarpıldıktan sonra en küçüğü yapay sinir hücresinin net girdisi olarak kabul edilir.
Çoğunluk	$\text{Net Girdi} = \sum_{i=1}^N \text{sgn}(w_i x_i)$	Girdiler ağırlıklar ile çarpıldıktan sonra pozitif ve negatif olanların sayısı bulunur. Büyük olan sayı hücrenin net girdisi olarak kabul edilir.

2.2.3.4. Aktivasyon Fonksiyonu

Transfer, eşik veya işaret fonksiyonları olarak da adlandırılan aktivasyon fonksiyonları, hücrenin davranışını belirleyen önemli unsurlardan biridir. Hücreye gelen net girdiyi işleyerek hücrenin bu girdiye karşılık üreteceği çıktının belirlenmesini sağlar. Diğer bir ifadeyle hücrenin çıkış genliğini, istenilen değerler arasında sınırlar. Bu değerler çoğunlukla [-1, 1] veya [0, 1] arasındadır (Öztemel, 2006). Bilinen bazı aktivasyon fonksiyonları:

- **Sigmoid Aktivasyon Fonksiyonu;**

Değişim aralığı [0, 1] olan ve bu aralıkta doğrusal olmayan bir değişim gösteren bu fonksiyon için giriş – çıkış ifadesi Denklem (2.1)' de değişim grafiği de Şekil 2.3. (a)' da verilmiştir. YSA uygulamalarında en çok kullanılan aktivasyon fonksiyonudur (Öztemel, 2006).

$$f(\text{NetGirdi}) = \frac{1}{1 + e^{-\text{NetGirdi}}} \quad (2.1)$$

- **Hiperbolik Tanjant Aktivasyon Fonksiyonu;**

YSA uygulamalarında en çok kullanılan aktivasyon fonksiyonlarından biri olan bu fonksiyonun değişim aralığı [-1, 1]' dir. Fonksiyonun tanımı Denklem (2.2)' de, giriş – çıkış grafiği Şekil 2.3. (b)' de verilmiştir. Fonksiyon girişine bağlı olarak belirtilen aralıkta doğrusal olmayan bir değişim gösterir (Kakıcı, 2009).

$$\tanh(\text{NetGirdi}) = \frac{e^{\text{NetGirdi}} - e^{-\text{NetGirdi}}}{e^{\text{NetGirdi}} + e^{-\text{NetGirdi}}} \quad (2.2)$$

- **Doğrusal (Lineer) Aktivasyon Fonksiyonu;**

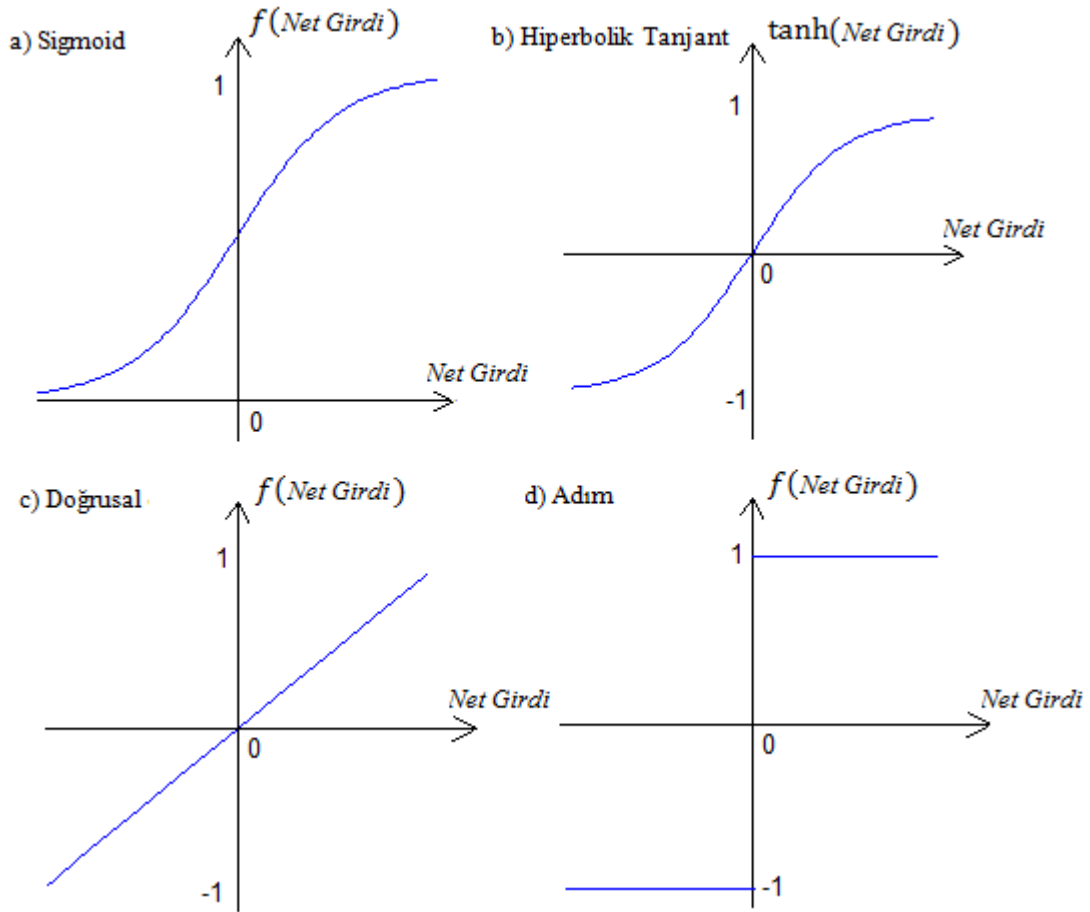
Fonksiyonun giriş – çıkış ifadesi ve davranışı sırasıyla Denklem (2.3) ve Şekil 2.3. (c)' de verilmiştir. Fonksiyonun girişine göre çıkışı doğrusal olarak değişmektedir. Denklem (2.3)' te A, sabit bir sayıdır (Kakıcı, 2009).

$$f(\text{NetGirdi}) = A \cdot \text{NetGirdi} \quad (2.3)$$

- **Adım (Step) Aktivasyon Fonksiyonu;**

Bu fonksiyonun matematiksel ifadesi ve grafiği sırasıyla Denklem (2.4) ve Şekil 2.3. (d)' de verilmiştir (Kakıcı, 2009).

$$f(\text{NetGirdi}) = \begin{cases} 1, & \text{NetGirdi} > 0 \\ -1, & \text{NetGirdi} \leq 0 \end{cases} \quad (2.4)$$



Şekil 2.4. Aktivasyon fonksiyonları

2.2.3.5. Çıktılar

Aktivasyon fonksiyonları tarafından belirlenen çıktı değerleridir. Üretilen çıktı ya dış dünyaya, başka bir hücreye ya da kendisine girdi olarak gönderilebilir (Öztemel, 2006).

2.2.4. Yapay Sinir Ağlarının Sınıflandırılması

Yapay sinir ağları işleyiş olarak her ne kadar birbirlerine benzese de tasarım ve işleyiş standartları yoktur (Kakıcı, 2009). Yapay sinir ağlarını, sinir hücreleri dizilimlerine ve bu hücreler arasındaki ağırlıkları düzenlemek için kullanılan öğrenme algoritmalarına göre iki ayrı dalda inceleyebiliriz:

2.2.4.1. Yapılarına Göre Yapay Sinir Ağları

YSA, bünyesindeki hücrelerin birbirlerine bağlantı şekillerine göre ileri ve geri beslemeli olarak ikiye ayrılır:

- **İleri Beslemeli Yapay Sinir Ağları:** Hücreler, girişten çıkışa doğru düzenli katmanlar şeklindedir. Bir katmandan sadece kendinden sonraki katmanlar arasında bağlantı bulunmaktadır. Ağa gelen bilgiler giriş katmanına daha sonra sırasıyla gizli katmanlardan ve çıkış katmanından işlenerek geçer ve sonra dış dünyaya çıkar (Kakıcı, 2009). En geniş kullanıma sahip sinir ağlarıdır. MLP ve LVQ ağları örnek olarak verilebilir.

- **Geri Beslemeli Yapay Sinir Ağları:** İleri beslemeli ağların aksine bir hücrenin çıktısı sadece kendinden sonra gelen katmana girdi olarak verilmez. Kendinden önceki katmanda veya kendi katmanında bulunan herhangi bir hücreye girdi olarak verilebilir (Kakıcı, 2009). Elman ve Jordan ağları, bu ağlara örnek olarak gösterilebilir.

2.2.4.2. Öğrenme Algoritmalarına Göre Yapay Sinir Ağları

YSA' nın verilen girdilere göre istenen çıktıyı üretebilmesinin yolu, ağın öğrenebilmesinden geçer. Bu da hücreler arasındaki ağırlıkların değiştirilmesi ile yapılır. Yapay sinir ağları, öğrenme algoritmalarına göre danışmanlı, danışmansız ve destekleyici öğrenme olmak üzere üçe ayrılır (Öztemel, 2006):

- **Danışmanlı Öğrenme:** YSA, kullanılmaya başlanmadan önce eğitilir. Eğitim sırasında, hem girdi değerleri hem de o girdi değerleri için istenen çıktı değerleri sisteme verilir. Ağ kendisine gösterilen örneklerden genellemeler yaparak bir çözüm uzayı elde eder. Daha sonra bu çözüm uzayı, ağa verilen benzer girdiler için sonuç üretir. Eğitim aşaması genelde çok zaman alır. Danışmanlı öğrenme için MLP, geri yayılım ve delta kuralı örnek olarak verilebilir.

- **Danışmansız Öğrenme:** Sistemin öğrenmesine yardımcı olan herhangi bir öğretmen veya danışman yoktur. Sisteme sadece girdiler verilir, örneklerdeki parametreler arasındaki ilişkileri sistemin kendi kendine öğrenmesi beklenir. Danışmanlı öğrenmeye göre çok daha hızlıdır ve matematik algoritmaları daha basittir. ART (Adaptif Rezonans Teori) ağları örnek olarak verilebilir.

- **Destekleyici Öğrenme:** Bir danışman veya öğretmen, her bir girdi için istenen çıktıyı sisteme vermek yerine, sistemin kendisine verilen girdilere karşılık kendi

çıktısını üretmesini bekler ve üretilen çıktının iyi veya kötü olup olmadığını gösteren bir bilgi verir. Ağ bu bilgiyi dikkate alarak eğitim sürecini devam ettirir. Destekleyici öğrenmeye, LVQ ağı örnek gösterilebilir.

2.3. Sezgisel Algoritmalar

Sezgisel algoritmalar, herhangi bir amacı gerçekleştirmek veya hedefe varmak için alternatifler arasından etkili olanlara karar vermek amacıyla tanımlanan kriterler veya bilgisayar metotlarıdır. Bu tür algoritmalar, yakınsama özelliğine sahiptir ve kesin çözümü değil kesin çözüme yakın en iyi çözümü garanti edebilmektedirler. Sezgisel algoritmaların gerekliliği şu şekilde açıklanmaktadır (Karaboğa, 2011):

- Optimizasyon problemi, kesin çözümü bulma işleminin tanımlanamadığı bir yapıya sahip olabilir.
- Anlaşılabilirlik açısından sezgisel algoritmalar karar verici açısından çok daha basit olabilir.
- Sezgisel algoritmalar, öğrenme amaçlı ve kesin çözümü bulma işleminin bir parçası olarak kullanılabilir.
- Matematik formülleriyle yapılan tanımlamalarda genellikle gerçek dünya problemlerinin en zor tarafları (hangi amaçlar ve hangi sınırlamalar kullanılmalı, hangi alternatifler test edilmeli, problem verisi nasıl toplanmalı) ihmal edilir. Model parametrelerini belirleme aşamasında kullanılan verinin hatalı olması, sezgisel yaklaşımın üretebileceği alt optimal çözümden daha büyük hatalara sebep olabilir.

2.3.1. Parçacık Sürüsü Optimizasyonu

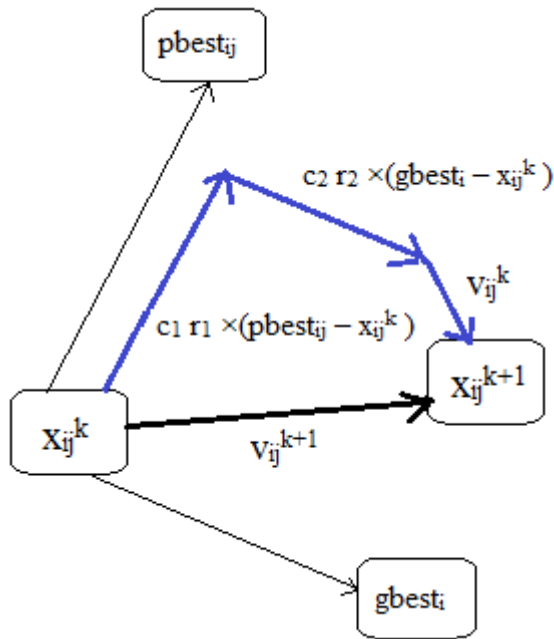
Popülasyon temelli sezgisel yöntemlerden biri olan Parçacık Sürüsü Optimizasyonu (PSO), ilk olarak 1995 - 1996 yıllarında Kennedy ve Eberhart (1995) tarafından kuş veya balık sürülerinin yiyecek bulma sırasındaki davranışlarından esinlenilerek geliştirilmiştir.

Yöntemin esasını oluşturan kuş sürüsündeki her bir kuş arama uzayındaki bireysel bir çözüme karşılık gelir ve “parçacık” (particle) olarak isimlendirilir. Parçacıkların tümü, optimize edilen uygunluk fonksiyonu (fitness function) tarafından değerlendirilen uygunluk (veya kalite) değerlerine ve parçacıkların uçuşlarını (araştırmayı) yönlendiren hızlara (velocities) sahiptir. Sürü, d-boyutlu problem uzayı

boyunca mevcut en iyi parçacığı takip ederek uçuşunu ve araştırmasını devam ettirir (Bağış, 2009).

2.3.1.1. Parçacık Sürüsü Optimizasyonu Algoritması

PSO algoritması başlangıçta rastgele bir parçacıklar (çözümler) grubu şeklinde oluşturulur ve daha sonra jenerasyonlar boyunca gelişimini sürdürerek araştırmasına devam eder. Her bir iterasyonda, her bir parçacık (çözüm) iki en iyi değer tarafından değişime uğratılarak geliştirilir ve küresel en iyi çözüm noktasına bir adım daha yaklaşması sağlanır (Şekil 2.5.). Bunların ilki, her bir parçacığın sahip olduğu ve o ana kadar ulaştığı en iyi değeri ifade eden “pbest” değeridir. Diğeri ise, sürüdeki (popülasyondaki) o ana değin elde edilmiş değere sahip ve algoritma tarafından kılavuz olarak izlenmekte olan küresel en iyi değerdir (gbest) (Liu ve ark., 2008; Bağış, 2009).



Şekil 2.5. Parçacıklar için konum güncellemesi

Sürüdeki parçacıklar ve bunların hızları, sırasıyla Denklem (2.5) ve (2.6)'daki gibi d-boyutlu birer diziyle gösterilir. N, sürüdeki toplam parçacık sayısını belirtmektedir.

$$x_i = (x_{i1} \ x_{i2} \ x_{i3} \ \dots \ x_{id}) \quad i=1, 2, 3, \dots, N \quad (2.5)$$

$$v_i = (v_{i1} \ v_{i2} \ v_{i3} \ \dots \ v_{id}) \quad i=1, 2, 3, \dots, N \quad (2.6)$$

Buna göre d-boyutlu arama uzayında dolaşan N adet parçacığın konumları (2.7) ve hızları (2.8) aşağıdaki matrislerle ifade edilir:

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1d} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2d} \\ \dots & \dots & \dots & \dots & \dots \\ x_{N1} & x_{N2} & x_{N3} & \dots & x_{Nd} \end{bmatrix} \quad (2.7)$$

$$V = \begin{bmatrix} v_{11} & v_{12} & v_{13} & \dots & v_{1d} \\ v_{21} & v_{22} & v_{23} & \dots & v_{2d} \\ \dots & \dots & \dots & \dots & \dots \\ v_{N1} & v_{N2} & v_{N3} & \dots & v_{Nd} \end{bmatrix} \quad (2.8)$$

$$v_{ij}^{k+1} = v_{ij}^k + c_1 r_1 \times (pbest_{ij} - x_{ij}^k) + c_2 r_2 \times (gbest_i - x_{ij}^k) \quad (2.9)$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \quad (2.10)$$

Her bir parçacığın hızı ve konumu (2.9) ve (2.10) denklemleri yardımıyla güncellenir. Bu denklemlerde; k değeri iterasyon sayısını, r_1 ve r_2 değerleri $[0, 1]$ aralığında normal dağılımlı rastgele sayıları, c_1 ve c_2 değerleri öğrenme sabitlerini, v_{ij} değeri i. parçacığın j. boyuttaki hızını, x_{ij} değeri ise konumunu ifade etmektedir (Gözde ve ark., 2010).

$$pbest_{ij} = (pbest_{i1} \ pbest_{i2} \ pbest_{i3} \ \dots \ pbest_{id}) \quad i=1, 2, 3, \dots, N \quad (2.11)$$

$$Pbest = \begin{bmatrix} pbest_{11} & pbest_{12} & pbest_{13} & \dots & pbest_{1d} \\ pbest_{21} & pbest_{22} & pbest_{23} & \dots & pbest_{2d} \\ \dots & \dots & \dots & \dots & \dots \\ pbest_{N1} & pbest_{N2} & pbest_{N3} & \dots & pbest_{Nd} \end{bmatrix} \quad (2.12)$$

$$gbest = (gbest_1 \ gbest_2 \ gbest_3 \ \dots \ gbest_d) \quad i=1, 2, 3, \dots, N \quad (2.13)$$

Yerel en iyi konum, her bir parçacık için d-boyutlu bir dizi ile gösterilirken (2.11) tüm parçacıklar için $N \times d$ boyutlu bir matris (2.12) ile gösterilir. Küresel en iyi konum d-boyutlu bir dizi (2.13) ile gösterilir (Gözde ve ark., 2010).

Aşağıda PSO algoritması için gerekli olan prosedür özetlenmiştir.

For her parçacık için

Parçacığı başlangıç konuma getir

End

Do For her parçacık için

Uygunluk değerini hesapla

Eğer uygunluk değeri, pbest ten daha iyi ise;

Şimdiki değeri yeni pbest olarak ayarla

End

Tüm parçacıkların bulunduğu pbest değerlerinin en iyisini, tüm parçacıkların gbest'i olarak ayarla

For her parçacık için

(2.9) denkleme göre parçacık hızını hesapla

(2.10) denkleme göre parçacık pozisyonunu güncelle

End

While maksimum iterasyon sayısına ulaşana veya minimum hata koşulu sağlanana kadar devam et (Bağış ve Çetin, 2009)

2.3.1.2. Parçacık Sürüsü Optimizasyonu Parametre Kontrolü

Parçacık Sayısı (N): Kullanılacağı probleme göre değişmektedir. Doğal olarak basit problemlerde az sayıda parçacık kullanılabilirken, problemin boyutu arttıkça başka bir ifadeyle optimizasyon değişkenlerinin sayısı arttıkça gerekli olan parçacık sayısı da artacaktır (Yetkin, 2008).

Parçacık Boyutu (d): Parçacık boyutu kullanıldığı probleme göre değişmektedir (Tamer ve Karakuzu, 2006).

Parçacık Aralığı: Kullanıldığı probleme göre değişmekte olup farklı boyutlarda ve aralıklarda parçacıkları tanımlamak mümkündür (Tamer ve Karakuzu, 2006).

V_{max} : Hız için bir maksimum değer (V_{max}) kullanılmaması parçacıkların araştırma uzayı dışına çıkmalarına sebep olmaktadır. Eğer parçacığın hızı, bunu aşarsa Denklem (2.14)' teki gibi V_{max} değerine eşitlenir. V_{max} parametresi algoritmanın küresel ve yerel arama kabiliyetleri arasındaki denge açısından önemli bir faktördür. V_{max} için büyük bir değer kullanılması küresel arama kabiliyetini artırır, küçük bir V_{max} ise yerel aramayı kolaylaştırır. Öte yandan çok küçük bir V_{max} yerel optimum bir çözümün

elde edilmesine çok büyük bir V_{\max} ise parçacığın, önceki iyi bir çözümün etrafında salınım yapmasına neden olmaktadır (Yetkin, 2008; Eberhart ve Shi, 2001; Gallad ve ark., 2002; Parsopoulos ve Vrahatis, 2002).

$$v_{ij}^{k+1} = \begin{cases} v_{ij}^{k+1} & , V_{\max} > v_{ij}^{k+1} \\ V_{\max} & , V_{\max} \leq v_{ij}^{k+1} \end{cases} \quad (2.14)$$

Öğrenme Faktörleri (c_1, c_2): Sürüdeki her bir parçacığı pbest ve gbest pozisyonlarına doğru çeken, stokastik hızlanma terimlerini ifade eden pozitif değerli sabitlerdir. c_1 , parçacığın kendi tecrübelerine göre hareket etmesini, c_2 ise sürüdeki diğer parçacıkların tecrübelerine göre hareket etmesini sağlar. Düşük değerler seçilmesi parçacıkların hedef bölgeye doğru çekilmeden önce, bu bölgeden uzak yerlerde dolaşmalarına imkân verir. Ancak hedefe ulaşma süresi uzayabilir. Diğer yandan, yüksek değerler seçilmesi, hedefe ulaşmayı hızlandırırken, beklenmedik hareketlerin oluşmasına ve hedef bölgenin gözden kaçırılmasına sebep olabilir. Bu algoritma üzerinde araştırmacıların yaptığı denemelerde $c_1=c_2=2$ olarak almanın iyi sonuçlar verdiği belirtilmiştir (Bağış ve Çetin, 2009).

Durdurma Kriteri: Bir durdurma kriteri seçilirken iki önemli nokta dikkate alınmalıdır. Birincisi, durdurma şartı PSO' nun erken yakınsamasına sebep olmamalıdır. Aksi takdirde bu durum, yerel optimum çözümlerin bulunmasına sebep olacaktır. İkincisi, durdurma kriteri uygunluk fonksiyonunun yoğun şekilde hesaplanmasını gerektirirse araştırma işleminin hesaplama karmaşıklığı artacaktır. Maksimum iterasyon sayısına (n) erişilmişse, kabul edilebilir bir çözüm elde edilmişse, belirli bir adım sayısınca gbest değeri değişmemişse veya amaç fonksiyonunun eğimi sifıra yaklaşmışsa algoritmanın çalışması durdurulabilir (Karaboğa, 2011).

Eylemsizlik Ağırlığı (w): Eski hız bilgisinin yeni hız bilgisine etkisini ayarlayan parametredir. Büyük eylemsizlik ağırlığı küresel arama, küçük ağırlık ise yerel arama yapılmasını kolaylaştırır. Eylemsizlik ağırlığı (inertia weight), yerel ve küresel arama arasındaki dengeyi sağlar ve bunun sonucunda yeterli optimal sonuca daha az iterasyonda ulaşılır. Buradaki her parçacık; sürüdeki sadece en iyi parçacığın değil sürüdeki diğer tüm parçacıkların tecrübelerinden de yararlanmış olur (Tamer ve Karakuzu, 2006). Eylemsizlik ağırlığının kullanılmasıyla (2.9)' daki hız değişim denklemi Denklem (2.15) haline dönüşür. Denklem (2.16)' da eylemsizlik ağırlığının

iterasyonla deęişimine dayanan bir yöntem verilmiştir, bu denklemde yer alan n deęeri maksimum iterasyon sayısını ifade etmektedir..

$$v_{ij}^{k+1} = w^k \times v_{ij}^k + c_1 r_1 \times (pbest_{ij} - x_{ij}^k) + c_2 r_2 \times (gbest_i - x_{ij}^k) \quad (2.15)$$

$$w^k = w_{max} - k \times \frac{(w_{max} - w_{min})}{n} \quad (2.16)$$

Azaltma Katsayısı (t): Eylemsizlik aęırlık deęerinin doęrusal olarak azaltılması için kullanılır. Azaltma katsayısının w deęerinin güncellenmesinde kullanımı, Denklem (2.17)' de verilmiştir:

$$w^{k+1} = w^k \times t \quad (2.17)$$

Sınırlama Katsayısı (χ): Verilen sınırlar altında hedefe yakınsamayı saęlar. Küçük deęerler seçildiğinde, algoritmanın küresel arama kabiliyeti düşer, yerel arama kabiliyeti artar; büyük deęerler seçildiğinde tam tersi bir durum oluşturur. χ deęerinin hesaplanması, Denklem (2.18)' e göre yapılır. Bu denklemde yer alan ϕ katsayısının hesaplanmasında (2.19) ve (2.21) denklemlerinden biri kullanılır. (2.19) denkleminin tercih edilmesi durumunda Denklem (2.20), (2.21) denkleminin tercih edilmesi durumunda ise Denklem (2.22) hız güncellemesi için kullanılır. Bu denklemler $\phi \geq 4$ şartı altında kullanılır (Karaboęa, 2011). (2.18) ve (2.19) denklemlerine göre yapılır.

$$\chi = \frac{2}{|2 - \phi - \sqrt{\phi(\phi - 4)}|} \quad (2.18)$$

$$\phi = c_1 + c_2 \quad (2.19)$$

$$v_{ij}^{k+1} = \chi [v_{ij}^k + c_1 \times r_1 \times (pbest_{ij} - x_{ij}^k) + c_2 \times r_2 \times (gbest_i - x_{ij}^k)] \quad (2.20)$$

$$\phi_1 = c_1 \times r_1, \phi_2 = c_2 \times r_2 \rightarrow \phi = \phi_1 + \phi_2 \quad (2.21)$$

$$v_{ij}^{k+1} = \chi [v_{ij}^k + \phi_1 \times (pbest_{ij} - x_{ij}^k) + \phi_2 \times (gbest_i - x_{ij}^k)] \quad (2.22)$$

Salınım Matrisi (A): Parçacıkların hız güncellemesi yapılırken çok küçük deęişimler olması için kullanılır. A matrisi (0, 1) arasında rastgele sayılardan oluşmaktadır. (2.23)' teki A matrisinin (2.20) ve (2.22) hız denklemlerine eklenmesi sonucunda, sırası ile Denklem (2.24) ve Denklem (2.25) elde edilmiştir. Bu

denklemlerde yer alan α katsayısı, çok küçük sabit bir katsayıdır. Uygulama yapılırken değeri 10^{-5} olarak alınmıştır.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1d} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2d} \\ \dots & \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & a_{N3} & \dots & a_{Nd} \end{bmatrix} \quad (2.23)$$

$$v_{ij}^{k+1} = \chi [v_{ij}^k + c_1 \times r_1 \times (pbest_{ij} - x_{ij}^k) + c_2 \times r_2 \times (gbest_i - x_{ij}^k) + \alpha \times a_{ij}] \quad (2.24)$$

$$v_{ij}^{k+1} = \chi [v_{ij}^k + \varphi_1 \times (pbest_{ij} - x_{ij}^k) + \varphi_2 \times (gbest_i - x_{ij}^k) + \alpha \times a_{ij}] \quad (2.25)$$

3. YAPAY SİNİR AĞLARININ EĞİTİMİ

YSA, olayların örneklerine bakarak onlardan ilgili olay hakkında genellemeler yapar, bilgi toplar, kendi deneyimlerini oluşturur ve daha sonra önceden hiç görmediği örnekler ile karşılaşınca öğrendiği bu bilgileri kullanarak o örnekler hakkında karar verebilmektedir (Öztemel, 2006).

YSA' nın eğitimi ve test edilmesi için epilepsi ve epilepsi hastası olmayan kişilerden alınan EEG verileri kullanılmıştır. Dolayısı ile giriş ve çıkış değerleri eğitim için birlikte kullanıldığından danışmalı öğrenme gerçekleştirilmiştir. EEG verileri, eğitim ve test işlemleri için iki kısma ayrılmıştır.

Öğrenme algoritması olarak temelde iki farklı metot seçilmiştir: Birincisi günümüzde en yaygın kullanıma sahip danışmalı öğrenme yöntemlerinden biri olan geri yayılım (BP) algoritması, ikincisi ise sezgisel optimizasyon yöntemlerinden biri olan parçacık sürüsü optimizasyonudur. YSA' nın PSO ile eğitiminde, eylemsizlik ağırlığının kullanılıp kullanılmamasına göre, kullanıldığı durumlarda değerinin güncellenmesinde kullanılan yöntemlere göre PSO' lar çeşitlendirilmiştir.

3.1. Geri Yayılım Algoritması ile YSA' nın Eğitimi

YSA' nın eğitilmesinde kullanılan öğrenme algoritmalarından biri olan geri yayılım algoritması, ileri beslemeli ve çok katmanlı bir ağ yapısı gerektirmektedir. Çok katmanlı bir geri yayılım ağı; bir giriş katmanı, bir çıkış katmanı ve en az bir gizli katmandan oluşur.

Geri yayılım algoritması, danışmanlı öğrenme yöntemlerinden birisidir ve giriş katmanı – gizli katman, gizli katman – gizli katman ve gizli katman – çıkış katmanı arasındaki ağırlıkların, ağ çıkışında meydana gelen hataya bağlı olarak optimize edilmesini amaçlamaktadır. Giriş ve ilgili çıkış vektörü ağı eğitmek için kullanılır. Giriş vektörü, ağı giriş katmanına verildikten sonra ağırlıklar vasıtası ile aradaki gizli katmanlardan geçerek en sondaki çıkış katmanına ulaşır. Ağdaki her bir hücre, kendisinde sonlanan ağırlık vektörlerinin aritmetiksel toplamını aldıktan sonra sonucu, kullanılan aktivasyon fonksiyonuna bağlı olarak kendinden sonraki katmanın tüm hücrelerine aktarır. Katmanlar arasında sigmoid aktivasyon fonksiyonu kullanıldığı kabulü altında ağırlıkların yenilenmesi için kullanılan ifadeler şu şekildedir (Altun ve ark., 2002):

Giriş katmanına gelen girdiler herhangi bir işleme uğramaksızın gizli katmana ağırlıklar vasıtası ile iletilirler. Gizli katmana gelen net girdi toplama fonksiyonu yardımı ile hesaplanır, aktivasyon fonksiyonundan geçirilir. Böylece gizli katman çıkışları hesaplanır ve elde edilen çıkışlar çıkış katmanına iletilir. Denklem (3.1)' de, I giriş katmandaki hücre sayısı, S gizli katmandaki hücre sayısıdır. r giriş katmanındaki ilgili hücrenin indisi, c gizli katmandaki ilgili hücrenin indisi olmak üzere $r=1, 2, \dots, I$ ve $c=1, 2, \dots, S$ ' dir. x_r , giriş katmanındaki r.hücreden gizli katmandaki c.hücreye gelen girdiyi, w_{rc} ise bu iki hücre arasındaki ağırlığı, net_c gizli katmandaki c.hücreye gelen net girdiyi ifade etmektedir. Denklem (3.2)' de y_c , c.hücrenin çıkışıdır, hesaplanmasında sigmoid aktivasyon fonksiyonu kullanılmıştır. Bu fonksiyonun türevi ise Denklem (3.3)' te verilmiştir.

$$net_c = \sum_{r=1, c=1}^{I, S} (w_{rc} x_r) \quad (3.1)$$

$$f(net_c) = \frac{1}{1 + e^{-net_c}} = y_c \quad (3.2)$$

$$f'(net_c) = (y_c - y_c \times y_c) = y_c (1 - y_c) \quad (3.3)$$

Gizli katman çıkışları ağırlıklar vasıtası ile çıkış katmanına iletilirler. Denklem (3.3)' de, O çıkış katmandaki hücre sayısıdır. o çıkış katmanındaki ilgili hücrenin indisi olmak üzere $o=1, 2, \dots, O$ ' dur. w_{co} ise gizli katmandaki c.hücre ile çıkış katmanındaki o.hücre arasındaki ağırlığı, net_o çıkış katmanındaki o.hücreye gelen net girdiyi, Denklem (3.2)' de y_o ise o.hücrenin çıkışını ifade etmektedir (Öztemel, 2006).

$$net_o = \sum_{c=1, o=1}^{S, O} (w_{co} y_c) \quad (3.3)$$

$$y_o = \frac{1}{1 + e^{-net_o}} \quad (3.4)$$

Geri yayılım algoritmasında kullanılan karesel ortalama hata (MSE) hesabı Denklem (3.5)' te verilmiştir. Burada P değeri, toplam örnek sayısını $p=1, 2, 3, \dots, P$ olmak üzere örnek indisini ifade etmektedir. Gerçek çıkış değeri y_{op} olan p.örneğin ağa gösterilmesi ile elde edilen çıkış değeri y_p' dir. Elde edilen hatanın geri yayılması ile

ağırlıklar güncellenir. Ağırlıkların güncellenmesinde delta kuralı kullanılır. Gizli – çıkış katman ağırlıklarının güncellenmesinde (3.6), (3.7) ve (3.10) denklemleri kullanılır. Giriş – gizli katman ağırlıklarının güncellenmesi ise (3.8), (3.9) ve (3.11) denklemleri ile yapılır (Batar, 2005; Sezer, 2008). λ , öğrenme katsayısı ve δ ise ilgili hücrenin eğimidir.

$$MSE = \text{hata} = \frac{1}{2P} \sum_{p=1}^P (y_p - y_{o_p})^2 \quad (3.5)$$

$$\delta_c = \text{hata} \times y_o (1 - y_o) \quad (3.6)$$

$$\Delta w_{co} = \lambda \times \delta_c \times y_c \quad (3.7)$$

$$\delta_r = w_{co} \times \delta_c \times y_c (1 - y_c) \quad (3.8)$$

$$\Delta w_{rc} = \lambda \times \delta_r \times x_r \quad (3.9)$$

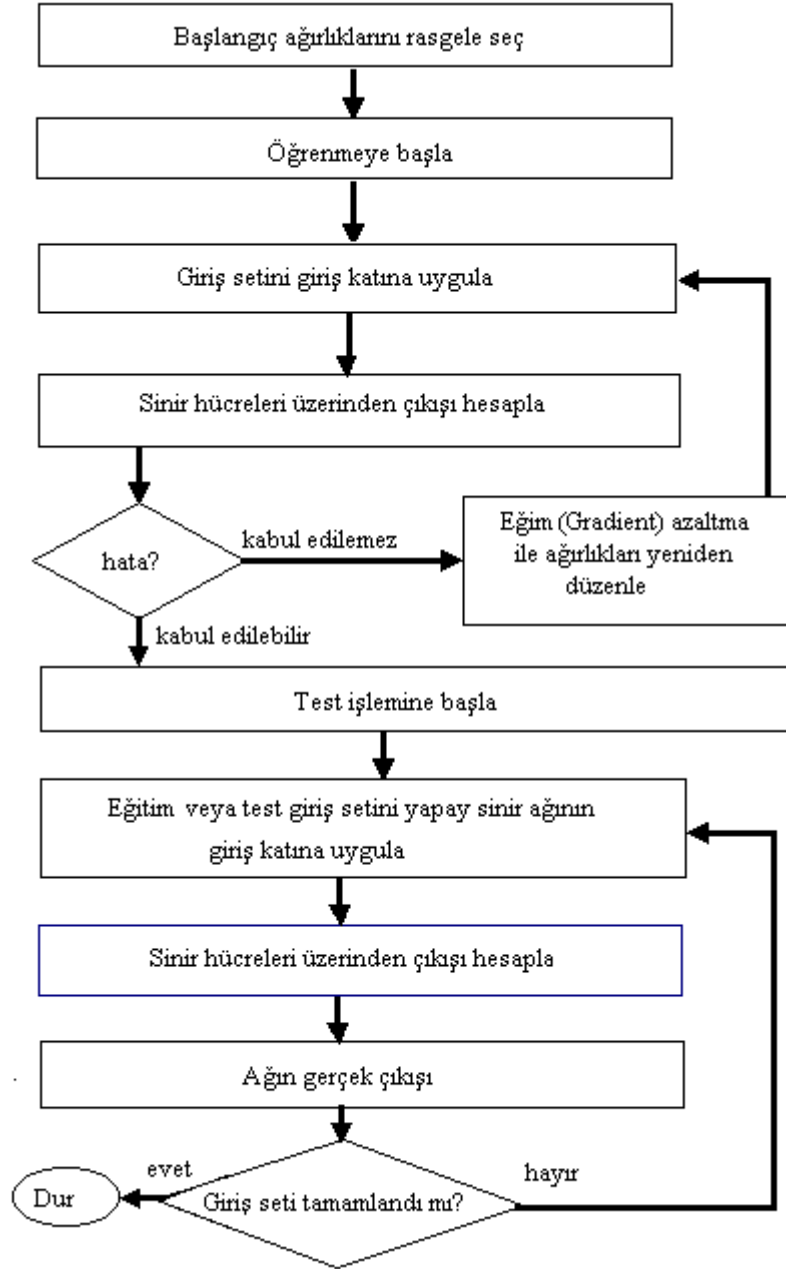
$$w_{co}(n+1) = w_{co}(n) + \Delta w_{co} \quad (3.10)$$

$$w_{rc}(n+1) = w_{rc}(n) + \Delta w_{rc} \quad (3.11)$$

Öğrenme Katsayısı(λ): (3.7) ve (3.9) denklemlerinde kullanılan öğrenme katsayısı, ağırlıkların bir sonraki adımda hangi oranda değiştirileceğini gösterir. Öğrenme katsayısı küçük seçildiğinde, ağın sonuca ulaşması yavaşlar, büyük seçildiğinde ise ağ daha kısa sürede sonuca ulaşır. Ancak çok yüksek değer seçilmesi hesaplamalarda büyük salınımlara neden olur ve ağın optimum noktayı bulmasını engelleyebilir (Kulluk, 2009).

Delta Kuralı: En çok kullanılan öğrenme algoritmalarından birisidir. Beklenen çıktı ile ağın çıktısı arasındaki farkı (hata) azaltmak için hücreler arasındaki bağlantı değerlerinin (ağırlıkların) iterasyonla değiştirilmesi ilkesine dayanarak geliştirilmiştir. Ortalama karesel hatayı (MSE) en aza indirmek amaçlanmaktadır. Hata, çıkış katmanından giriş katmanına geri iletilerek azaltılmaktadır (Kulluk, 2009).

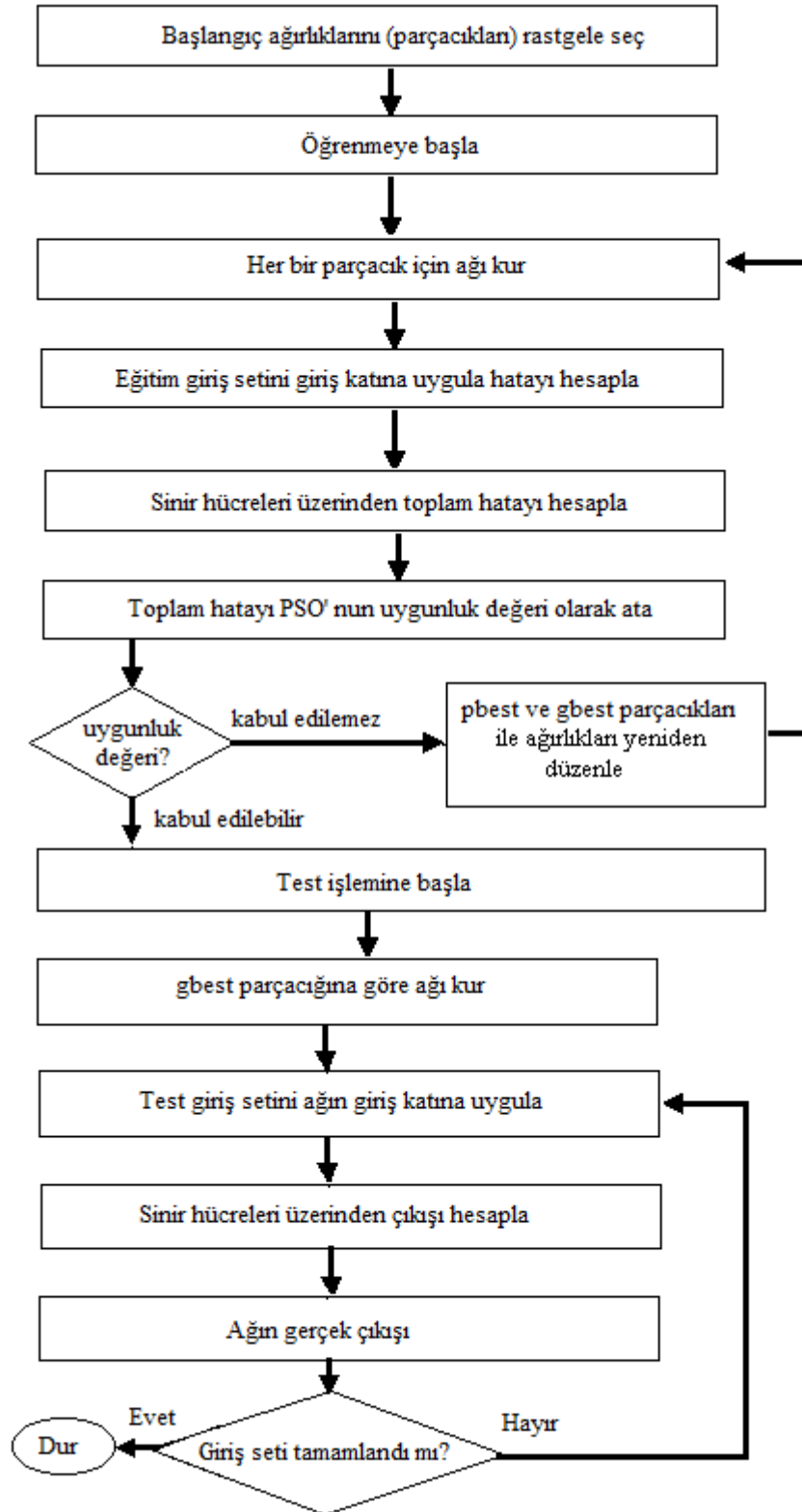
Delta öğrenme kuralını kullanan geri yayılım algoritmasının akış şeması Şekil 3.2. (Sağiroğlu ve ark., 2003)' de verilmiştir.



Şekil 3.1. Geri yayılım algoritması akış diyagramı

3.2. PSO ile YSA' nın Eğitimi

YSA' da öğrenme, katmanlar arası ağırlıkların en uygun duruma getirilmesi sürecidir ve bu bir optimizasyon işlemidir. Bu yüzden günümüzde kullanımı yaygınlaşmakta olan PSO, YSA' nın eğitimi için tercih edilmiştir.



Şekil 3.2. PSO ile YSA'nın eğitilmesinin akış diyagramı

PSO ile YSA' nın eğitilmesi ve test edilmesi işlemlerinin yer aldığı akış şeması Şekil 3.2.' de verilmiştir. YSA' nın öğrenme işlemine, katmanlar arasındaki bağlantıların sayısal değerlerini tutan ağırlıkların rastgele oluşturulması ile başlanır. Bu ağırlıklar, PSO için parçacık değerlerini ifade etmektedir. Katmanlar arasındaki bağlantı sayısı ise parçacığın boyutunu ifade etmektedir.

Her bir parçacığa göre ağ kurulur ve eğitim örnekleri sırası ile ağa gönderilir. Örneğin ağa sunulmasından sonra, çıkış olarak elde edilen değer ile gerçekte olması gereken değer arasındaki fark (hata) hesaplanır. Bütün örnekler ağa sunulduktan sonra toplam hata (MSE) hesaplanır ve elde edilen değer parçacığın uygunluk değeri olarak kabul edilir. İlk adımda bu uygunluk değeri, parçacığın pbest değeri; parçacıklar içerisindeki en iyi uygunluk değeri ise gbest değeri olarak atanır.

Uygunluk değeri yani hata kabul edilebilir düzeyde değil ise pbest ve gbest değerleri ile parçacıkları güncellenir. Yeni parçacık değerlerine göre yeniden ağ kurulur, yeniden örnekler ağa verilir ve uygunluk değer hesabı yapılır. Bu işlemler, o zamana kadar elde edilmiş en iyi uygunluk değeri (gbest), istenilen (0'a çok yakın) değere veya maksimum iterasyona kadar ulaşıncaya devam eder.

Hata kabul edilebilir düzeyde ise test işlemine geçilir. Bu sefer ağ gbest parçacık değerlerine göre kurulur, test örnekleri sırası ile ağın giriş katmanına gönderilir ve elde edilen değerler örneğin çıkışı olarak verilir.

En son elde edilen gbest değeri, ağın çıkışına herhangi bir eşik uygulanmadığı durumda ağın sınıflandırma performansını vermektedir.

4. UYGULAMA

Bu tez çalışmasında YSA' nın eğitimi için epileptik ya da hasta olmayan kişilere ait EEG verileri kullanılmıştır. 800'ü epileptik kişilere ait olmak üzere 1600 sayıda kişiden alınan veri, içerisinde EEG sinyalinin alınan 512 adet örneği barındırmaktadır. Bunlardan 1200 adeti eğitim için, kalan 400 adeti test için ayrılmıştır. Eğitim ve test verilerinde epileptik ve hasta kişilere ait verilerin bulunma oranı birbirine eşittir. 512 adet örneğin minimum, maksimum, ortalama ve standart sapması alınarak değer sayısı dörde indirgenmiştir. Daha sonra bütün veriler [0, 1]' da normalize edilmiştir, bunun içinde standart bir formül olan Denklem (4.1) kullanılmıştır.

$$veri_i = \frac{veri_i - veri_{min}}{veri_{max} - veri_{min}} \quad (4.1)$$

Elde edilen veriler çeşitli PSO versiyonlarından biri veya geri yayılım algoritması kullanılarak eğitilen sinir ağlarının girişlerine verilmiştir. Geri yayılım algoritmasını kullanan ağ için gizli katmandaki hücre sayısı değiştirilerek ve PSO ile eğitilen ağlar için hücre sayısında ek olarak parçacık sayısı ve epok sayısı da değiştirilerek denemeler yapılmış, elde edilen sonuçlar karşılaştırılmıştır.

Sınıflandırma işleminin doğruluğunun belirlenebilmesinde duyarlılık (sensitivity) ve belirlilik (specificity) kavramlarından yararlanılır. Bu kavramların hesaplanmalarında (4.2) ve (4.3) denklemleri kullanılır. Bu denklemlerde geçen parametrelerin tanımları (Metz, 1978; Sezer, 2008);

- TP (True Positive): Hasta kişiye hasta teşhisi konması,
- TN (True Negative): Hasta olmayan kişiye hasta değil teşhisi konması,
- FP (False Positive): Hasta olmayan kişiye hasta teşhisi konması,
- FN (False Negative): Hasta olan kişiye hasta değil teşhisi konması durum sayılarıdır.

$$Duyarlılık = \frac{TP}{TP + FN} \quad (4.2)$$

$$Belirlilik = \frac{TN}{TN + FP} \quad (4.3)$$

Duyarlılık, hasta kişilere ait verilerin; belirlilik ise hasta olmayan kişilere ait verilerin tahmin yüzdesidir (Özbeyaz, 2010). Duyarlılık ve belirlilik analizleri yapıldıktan sonra hangi PSO yönteminin en iyi sınıflandırma doğruluğunu verdiğini belirlemek amacıyla 30 ve 40 parçacık için her bir eğitim işlemi önce 10, sonra 30 parçacık için 30 kez koşturulmuştur. Bunlar içerisinde en iyi (best) ve en kötü (worst) olan uygunluklar bulunmuş, ortalama uygunluk değerleri ile birlikte bu değerlerin değişimi grafikte sunulmuştur.

Eğitim ve test doğruluk oranlarının belirlenmesinde eşik değeri 0,4 olarak seçilmiştir. Sınıflandırma sonucu, bu değer üstünde kalan çıkış değerinin hasta ve altında kalan çıkış değerinin sağlıklı kişiler olacak şekilde doğru sınıflandırıldığı kabul edilmiştir. Ayrıca, sinir ağlarının gizli ve çıkış katmanına eşik uygulanmış ve eşik değeri 1 seçilmiştir.

4.1. Geri Yayılım Algoritması ile YSA' nın Gerçeklenmesi

Bu çalışmada, 3 katmanlı bir MLP sinir ağı oluşturulmuş, öğrenme algoritması olarak geri yayılım algoritması kullanılmıştır. Elde edilen ağ BPNN (Geri Yayılımlı Sinir Ağı) olarak isimlendirilmiştir. Verilerdeki özellik sayısı dörde indirildiği için, giriş katmanında doğal olarak dört hücre bulunacaktır. Kişiler epileptik ve hasta olmayan şeklinde sınıflandırılacaktır. Kişi epileptik ise 1, değil ise 0 çıkışı alınacağından çıkış katmanında bir hücre bulunmalıdır. Katmanlardaki optimum hücre sayısını genellikle deneysel olarak bulunur. Bu nedenle gizli katmanındaki hücre sayısı sırasıyla 3, 4 ve 5 alınmış, hücre sayısındaki değişimin ağın performansına etkileri karşılaştırılmıştır. Hedef MSE değeri (durdurma kriteri) 0,001, öğrenme katsayısı 0,5 seçilmiştir.

Geri yayılım algoritmasında, hatanın geriye yayılması sürecinde türev alma işlemi yapıldığından türevi alınabilir bir aktivasyon fonksiyonunun kullanılması gerekir. Bu nedenle aktivasyon fonksiyonu olarak sigmoid fonksiyonu seçilmiştir. Böylece PSO ile eğitilmiş YSA ile BPNN' nin karşılaştırılabilmesi de mümkün olacaktır.

Gizli katman sayısının değiştirilmesi ile elde edilen doğruluk oranları, ulaşılan MSE değerleri Çizelge 4.1. ' de verilmiştir. Çizelge 4.1.' den görüldüğü gibi en iyi sonuç 5 gizli katman hücresi için bulunmuştur, ama diğer hücre sayılarının seçilmesi ile elde edilen sonuçlara bakıldığında aralarında pek fazla fark olmadığı, üstelik eğitim süresinin çok fazla arttığı ulaşılan iterasyon sayısından anlaşılmıştır. Ayrıca, Çizelge

4.1. ve Şekil 4.1. göz önünde tutulduğunda, daha az iterasyon sayısı ile istenilen değerlere ulaşılmıştır. Sadece hedef MSE değerine ulaşması daha fazla zaman almaktadır. Şekil 4.1., 4.2. ve 4.3. ve Çizelge 4.1.' de görüldüğü üzere gizli katman hücre sayısının artması eğitim ve sınıflandırma doğruluğunu küçük oranlarda artırmıştır. Bu nedenlerden dolayı, bu çalışmada gizli katman hücre sayısı 3 seçilmiştir.

Çizelge 4.1. Gizli katmandaki hücre sayısı değişimine göre BPNN' nin performansı

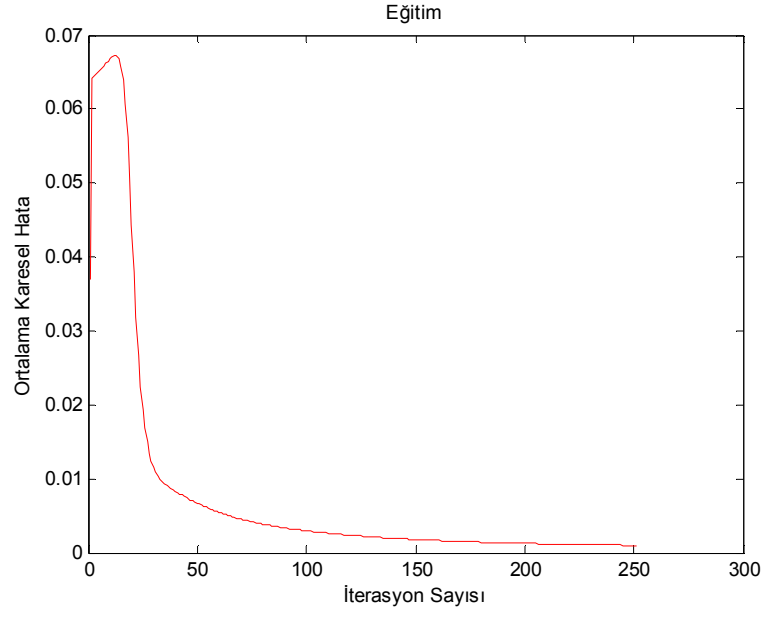
Hücre Sayısı	% Eğitim Doğruluk Oranı	% Test Doğruluk Oranı	MSE
3	99,8333	54,2500	0,001
4	99,8333	56	0,001
5	99,9167	58,7500	0,001

Gizli katman hücre sayısına göre eğitim ve test verileri olmak üzere iki kısımda hesaplanmasıyla elde edilen duyarlılık ve belirlilik analizi sonuçları Çizelge 4.2.' de verilmiştir.

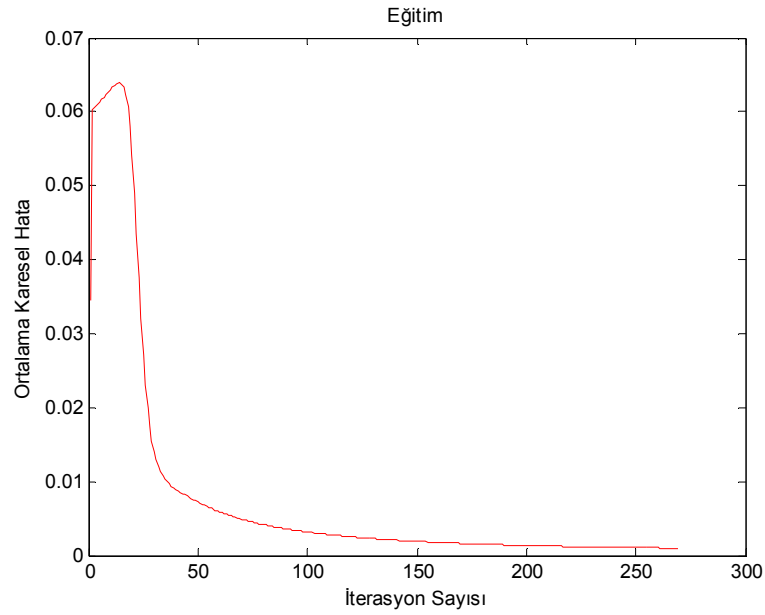
Çizelge 4.2. Gizli katmandaki hücre sayısı değişimine göre duyarlılık-belirlilik değerleri

Hücre Sayısı	Eğitim		Test	
	Duyarlılık	Belirlilik	Duyarlılık	Belirlilik
3	1	0,9967	1	0,0850
4	1	0,9967	1	0,1200
5	1	0,9983	1	0,1750

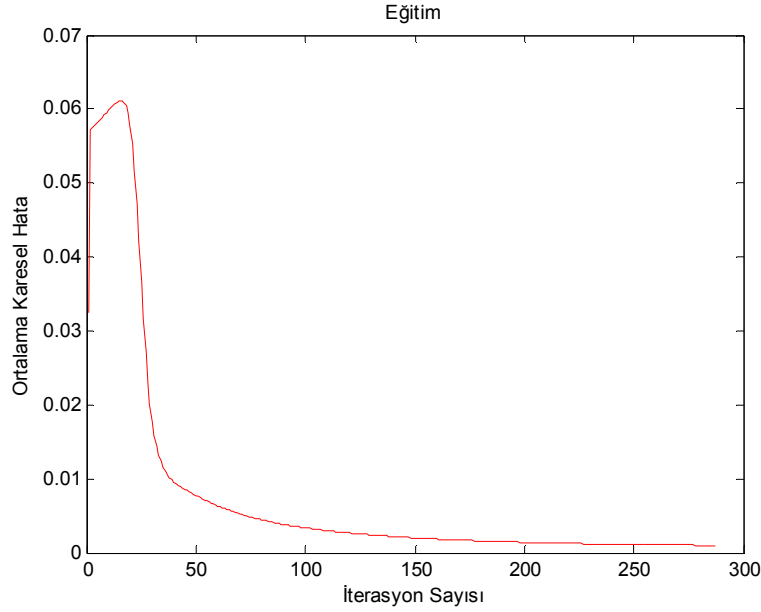
BPNN' nin gizli katmanındaki hücre sayısına bağlı olarak ortalama karesel hata değerlerinin iterasyonla değişimi ise Şekil 4.1., Şekil 4.2. ve Şekil 4.3.' de gösterilmiştir.



Şekil 4.1. 3 gizli katmanlı hücreli BPNN'nin iterasyon-MSE değişim grafiği



Şekil 4.2. 4 gizli katmanlı hücreli BPNN'nin iterasyon-MSE değişim grafiği



Şekil 4.3. 5 gizli katman hücreli BPNN' nin iterasyon-MSE değişim grafiği

4.2. Çeşitli PSO Yapıları ile YSA' nın Gerçeklenmesi

PSO algoritması, çözüm uzayını aramada üstün bir beceriye sahip olması (Delice, 2008) nedeniyle bu çalışmada kullanılmıştır.

YSA' nın katmanları arasındaki ağırlıklar, PSO' nun parçacık değerlerine karşılık gelmektedir. Dolayısıyla YSA' nın eğitimi sırasında gerçekleştirilen ağırlık güncellemeleri, parçacıkların hızına; ağırlıkların ağırlık değişim miktarları ile toplanmasıyla elde edilen yeni ağırlık değerleri ise parçacıkların hareket etmesi sonucu geçtikleri yeni konumlarını ifade etmektedir. Parçacık konumlarında meydana gelecek değişimler ağırlıklardaki değişimi ifade eder.

Hız güncelleme denkleminde yapılan değişimlere göre PSO yöntemi çeşitlendirilmiştir. Bu değişimler, eylemsizlik ağırlığı, sınırlama katsayısı ve salınım matrisi gibi değişkenlerin kullanımı, kullanılmaması ya da bu değişkenlerin hesaplanmasında farklı yöntemlerin tercih edilmesiyle gerçekleştirilmiştir. Bu PSO versiyonlarından birini kullanan sinir ağları, uygulamada kullanım kolaylığı açısından 7 farklı şekilde isimlendirilmiştir:

1. PS0NN1: Bu ağda, hız denkleminde eylemsizlik ağırlığı, sınırlama katsayısı, salınım matrisinden herhangi birinin yer almadığı PSO versiyonu

kullanılmıştır. Temel PSO' nun hız ve konum güncelleme denklemi sırasıyla Denklem (2.8) ve (2.9)' da verilmiştir.

2. PSNN2: Hız denklemi, (2.15)' tir. Bu denklemde yer alan eylemsizlik ağırlığının değerinin hesaplanmasında (2.16) denklemi kullanılmış ve w_{max} ve w_{min} değerleri sırasıyla 0,9 ve 0,4 seçilmiştir (Shi ve Eberhart, 1999; Delice, 2008; Wang ve ark., 2007).
3. PSNN3: Yine ağırlık güncellemesinde eylemsizlik ağırlığını kullanan PSO dikkate alınmış, ancak bu değişkenin iterasyonla değişiminde Denklem (2.17) kullanılmıştır. Bu denklemde kullanılan, azaltma katsayısının değeri 0,975, eylemsizlik ağırlığının başlangıç değeri ise 0,9 seçilmiştir (Şevkli ve Yenisey, 2006). Hız hesabı ise (2.15) denklemi ile yapılmıştır.
4. PSNN4: Sınırlama katsayısının yer aldığı (2.20) denklemi parçacıkların hızının güncellemesinde kullanılmıştır. Bu denklemde yer alan ϕ değişkeninin hesaplanmasında ise (2.19) denklemi tercih edilmiştir.
5. PSNN5: Ağırlıkların güncellemesinde salınım matrisini kullanan PSO tercih edilmiş ve Denklem (2.24)' e göre ağırlıklar güncellenmiştir.
6. PSNN6: Ağırlıkların güncellenmesinde sınırlama katsayısının yer aldığı (2.22) denklemi kullanan PSO dikkate alınmıştır, ancak belirtilen denklemde yer alan ϕ değişkeninin hesaplanması için (2.21) denklemi kullanılmıştır.
7. PSNN7: Ağırlık değerlerinin hesaplanmasında hareketlilik olması için salınım matrisi kullanılmış ve (2.25) hız denklemi tercih edilmiştir.

PSNN4, PSNN5, PSNN6 ve PSNN7 ağlarında PSO' nun hız denklemlerinde kullanılan, c_1 ve c_2 öğrenme faktörleri yapılan denemeler sonucunda birbirine eşit ve 2,1 alınmıştır. V_{max} değeri, yine yapılan denemeler sonucunda 0,1 seçilmiştir. Bu değerler ile hedefe daha fazla yakınsama sağlanmıştır.

Elde edilen ağlar 30 parçacık ve 3 gizli katman hücreli ağ için farklı epok (toplam iterasyon sayısı) sayısında çalıştırılarak eğitim ve test verilerindeki sınıflandırma doğrulukları elde edilmiş, en iyi uygunluk değerleri (Gbest) bulunmuş, duyarlılık ve belirlilik analizleri yapılmış, ağların uygunluk değerlerinin iterasyon ile değişim grafikler aracılığı ile gösterilmiştir. En uygun epok sayısı belirlenmeye çalışılmıştır.

Ağların toplamda 50 iterasyon çalıştırılması sonucunda elde edilen doğruluk oranları Çizelge 4.3.' te, duyarlılık ve belirlilik analizi sonuçları Çizelge 4.4.' te ve

iterasyon sayısı – uygunluk değeri değişim grafiği Şekil 4.4.’ te verilmiştir. Çizelge 4.3.’ te görüldüğü gibi en iyi sınıflandırmayı PSONN1, en kötü sınıflandırmayı ise PSONN7 gerçekleştirmiştir.

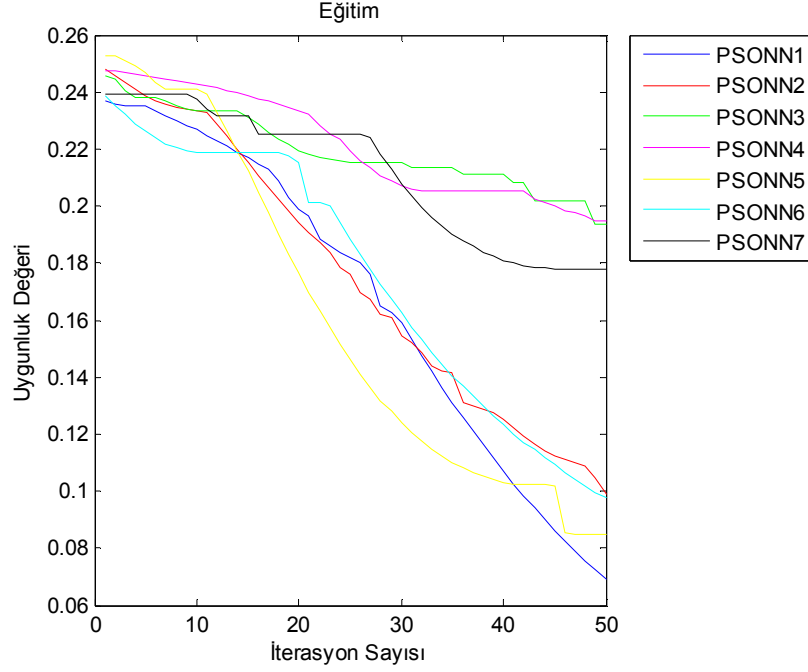
Çizelge 4.3. 50 epok için ağların performansı

Ağ Tipi	Gbest	% Eğitim Doğruluk Oranı	% Test Doğruluk Oranı
PSONN1	0,0692	93,5833	94,7500
PSONN2	0,0988	50	98,5000
PSONN3	0,1934	75,5000	78
PSONN4	0,1947	53	69,2500
PSONN5	0,0848	50	87,2500
PSONN6	0,0975	50	97,2500
PSONN7	0,1781	50	50

Çizelge 4.4. 50 epok için duyarlılık-belirlilik değerleri

Ağ Tipi	Eğitim		Test	
	Duyarlılık	Belirlilik	Duyarlılık	Belirlilik
PSONN1	0,8717	1	0,8950	1
PSONN2	0	1	1	0,9700
PSONN3	0,5100	1	0,5600	1
PSONN4	0,0600	1	0,3850	1
PSONN5	1	0	1	0,7450
PSONN6	0	1	0,9450	1
PSONN7	1	0	1	0

Rastgele 30 parçacık oluşturularak başlatılan ağların 50 epok çalıştırılması ile elde edilen sonuçlara bakıldığında PSONN1’ in daha iyi sınıflandırma yaptığı görülmüştür. Ancak genel bir yorum yapabilmek için 50 epok çalıştırmak yeterli değildir. Farklı epok sayılarında elde edilen sonuçları incelemek gerekir.



Şekil 4.4. 50 epok için iterasyon sayısı – uygunluk değeri değişim grafiği

Ağların 100 epok çalıştırılması sonucunda elde edilen doğruluk oranları, duyarlılık ve belirlilik analizi sonuçları ve iterasyon sayısı – uygunluk değeri değişim grafiği sırasıyla Çizelge 4.5., Çizelge 4.6. ve Şekil 4.5.’ te verilmiştir.

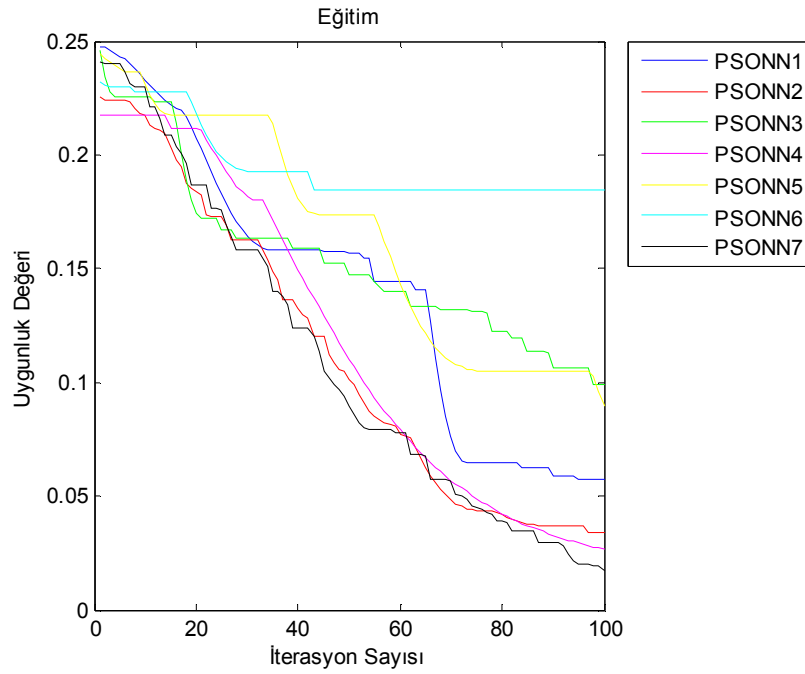
Sonuçlara bakıldığında en iyi sınıflandırmayı PSONN2’ nin yaptığı görülmüştür, ancak diğerleri istenileni verememiştir. Bu sebeple 100 epok çalıştırmanın yeterli olmadığı kanaatine varılmıştır.

Çizelge 4.5. 100 epok için ağların performansı

Ağ Tipi	Gbest	% Eğitim Doğruluk Oranı	% Test Doğruluk Oranı
PSONN1	0,0573	50,2500	99
PSONN2	0,0338	97,5833	100
PSONN3	0,0993	53,5833	98,7500
PSONN4	0,0266	50	99,2500
PSONN5	0,0899	50	90.5000
PSONN6	0,1844	67,9167	50
PSONN7	0,0169	50	100

Çizelge 4.6. 100 epok için duyarlılık-belirlilik değerleri

Ağ Tipi	Eğitim		Test	
	Duyarlılık	Belirlilik	Duyarlılık	Belirlilik
PSONN1	0,0050	1	1	0,9800
PSONN2	0,9517	1	1	1
PSONN3	0,0717	1	0,9750	1
PSONN4	1	0	0,9850	1
PSONN5	1	0	0,8100	1
PSONN6	0,3583	1	1	0
PSONN7	1	0	1	1



Şekil 4.5. 100 epok için iterasyon sayısı – uygunluk değeri değişim grafiği

Ağların 150 epok çalıştırılması sonucunda elde edilen doğruluk oranları, duyarlılık ve belirlilik analizi sonuçları ve iterasyon sayısı – uygunluk değeri değişim grafiği sırasıyla Çizelge 4.7., Çizelge 4.8. ve Şekil 4.6.' da verilmiştir.

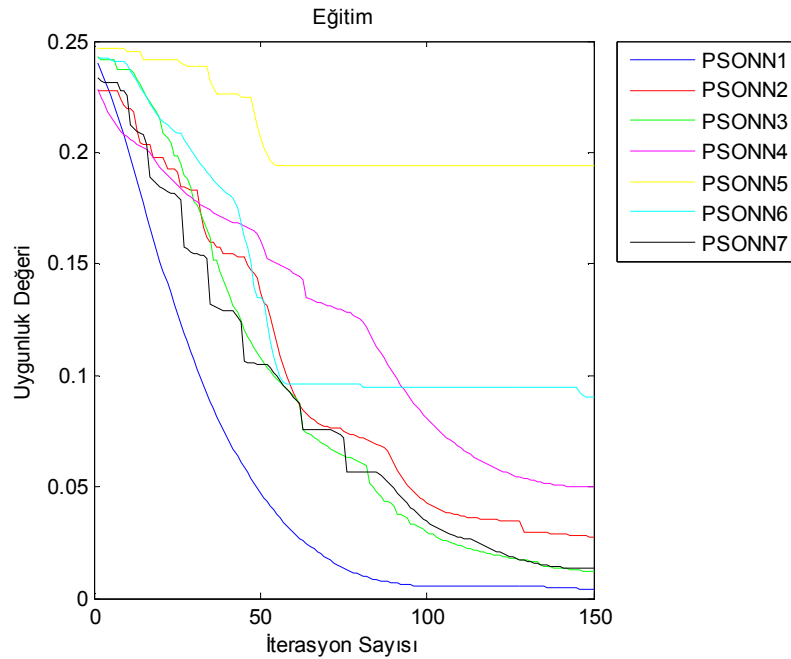
Çizelgelere bakıldığında eğitim verilerinde doğru sınıflandırmayı PSONN3' ün, test verilerinde ise PSONN1' in yaptığı görülmektedir.

Çizelge 4.7. 150 epok için ağların performansı

Ağ Tipi	Gbest	% Eğitim Doğruluk Oranı	% Test Doğruluk Oranı
PSONN1	0,0043	87,0833	99,7500
PSONN2	0,0275	50	97,2500
PSONN3	0,0120	96,4167	99,2500
PSONN4	0,0500	95	91
PSONN5	0,1940	53,9167	64,2500
PSONN6	0,0903	68,4167	99,2500
PSONN7	0,0136	50	99,5000

Çizelge 4.8. 150 epok için duyarlılık-belirlilik değerleri

Ağ Tipi	Eğitim		Test	
	Duyarlılık	Belirlilik	Duyarlılık	Belirlilik
PSONN1	0,7417	1	0,9950	1
PSONN2	1	0	0,9450	1
PSONN3	0,9283	1	0,9850	1
PSONN4	0,9000	1	1	0,8200
PSONN5	0,0783	1	0,9700	0,3150
PSONN6	0,3683	1	0,9850	1
PSONN7	1	0	1	0,9900



Şekil 4.6. 150 epok için iterasyon sayısı – uygunluk değeri değişim grafiği

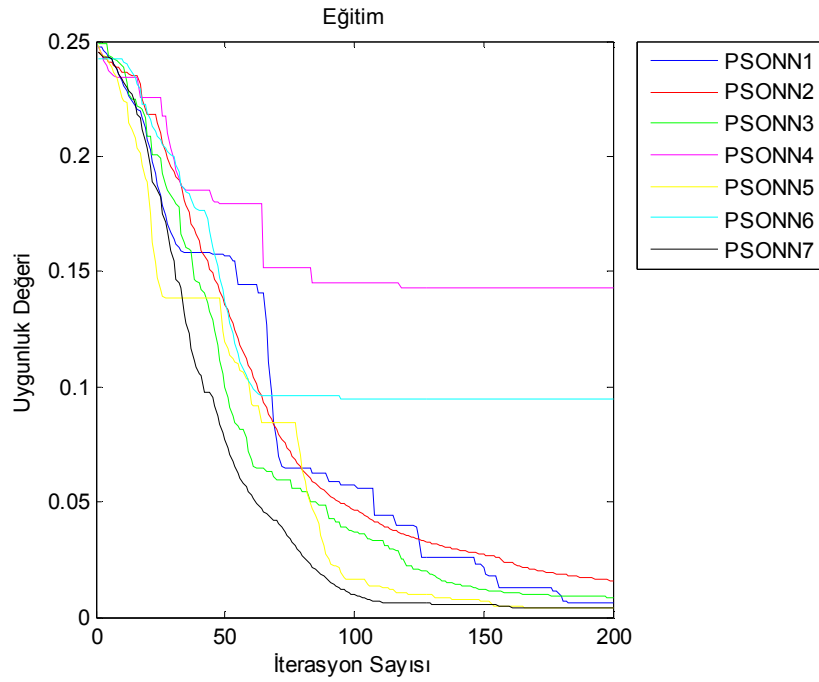
Ağların toplamda 200 iterasyon çalıştırılması sonucunda elde edilen doğruluk oranları Çizelge 4.9.' da, duyarlılık ve belirlilik analizi sonuçları Çizelge 4.10.' da ve iterasyon sayısı – uygunluk değeri değişim grafiği Şekil 4.7.' de verilmiştir. Hem eğitim hem de test verilerinde en iyi sınıflandırma oranlarını veren ağ, PSONN7' dir.

Çizelge 4.9. 200 epok için ağların performansı

Ağ Tipi	Gbest	% Eğitim Doğruluk Oranı	% Test Doğruluk Oranı
PSOENN1	0,0066	63	99,7500
PSOENN2	0,0159	90,5833	99,2500
PSOENN3	0,0087	99	99,2500
PSOENN4	0,1427	75,5000	89,2500
PSOENN5	0,0037	50	99,5000
PSOENN6	0,0947	50	88,7500
PSOENN7	0,0039	98,2500	100

Çizelge 4.10. 200 epok için duyarlılık-belirlilik değerleri

Ağ Tipi	Eğitim		Test	
	Duyarlılık	Belirlilik	Duyarlılık	Belirlilik
PSOENN1	0,2600	1	0,9900	1
PSOENN2	0,8117	1	0,9850	1
PSOENN3	0,9883	0,9917	0,9850	1
PSOENN4	0,6533	0,8567	0,8850	0,9000
PSOENN5	1	0	1	0,9900
PSOENN6	1	0	0,7750	1
PSOENN7	1	0,9650	1	1



Şekil 4.7. 200 epok için iterasyon sayısı – uygunluk değeri değişim grafiği

Ağların 250 epok çalıştırılması sonucunda elde edilen doğruluk oranları, duyarlılık ve belirlilik analizi sonuçları ve iterasyon sayısı – uygunluk değeri değişim grafiği sırasıyla Çizelge 4.11., Çizelge 4.12. ve Şekil 4.8.’ de verilmiştir. Eğitim ve test verilerinde sınıflandırma işlemini en iyi PSONN3’ ün gerçekleştirdiği görülmüştür.

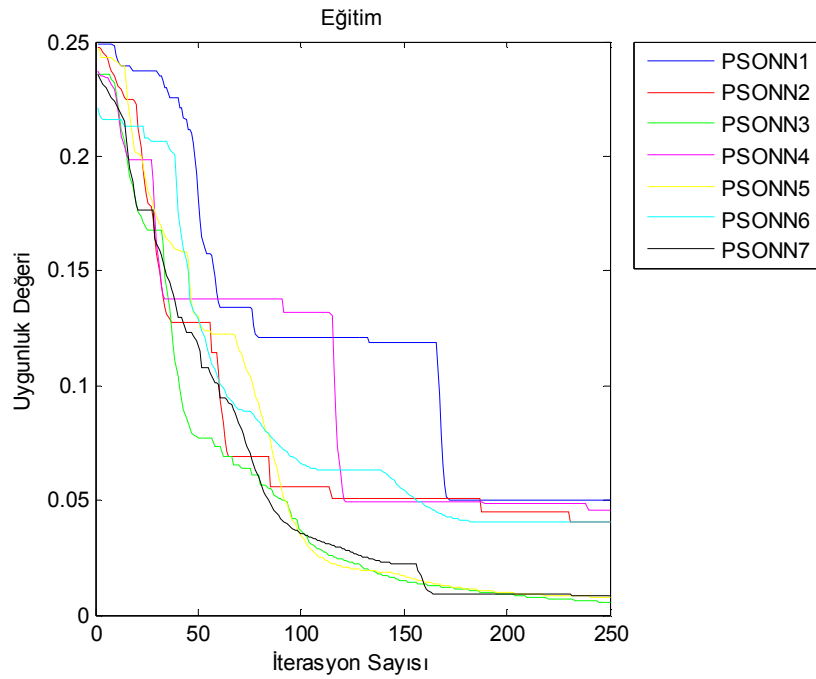
Görüldüğü üzere ağların farklı epok sayılarında verdiği gbest ve doğruluk oranları değerleri değişmiştir. Bunun sebebi, PSO yönteminin sezgisel bir yöntem olmasıdır. Ağların eğitiminde kullanılmak üzere parçacıklar başlangıçta rastgele oluşturulur, bu parçacıkların başlangıç değerlerinin optimal parçacık değerine yakın olup olmamasına göre sınıflandırma oranları değişmektedir. Bu sebeple her bir epok sayısında en iyi sınıflandırmayı gerçekleştiren ağ modelinin farklı olduğu görülmüştür.

Çizelge 4.11. 250 epok için ağların performansı

Ağ Tipi	Gbest	% Eğitim Doğruluk Oranı	% Test Doğruluk Oranı
PSONN1	0,0503	71,9167	97,5000
PSONN2	0,0408	88,9167	96
PSONN3	0,0056	99,4167	100
PSONN4	0,0453	62,6667	99
PSONN5	0,0079	50	100
PSONN6	0,0406	60,4167	96,2500
PSONN7	0,0088	76,4167	99

Çizelge 4.12. 250 epok için duyarlılık-belirlilik değerleri

Ağ Tipi	Eğitim		Test	
	Duyarlılık	Belirlilik	Duyarlılık	Belirlilik
PSOINN1	0,4383	1	0,9500	1
PSOINN2	0,7783	1	0,9200	1
PSOINN3	0,9950	0,9933	1	1
PSOINN4	0,2533	1	0,9800	1
PSOINN5	1	0	1	1
PSOINN6	0,2083	1	0,9250	1
PSOINN7	1	0,5283	1	0,9800



Şekil 4.8. 250 epok için iterasyon sayısı – uygunluk değeri değişim grafiği

Epok sayısı arttıkça hatanın genel olarak düştüğü gözlenmiştir. Ayrıca düşük epok sayısının seçilmesi işlem yükünü, kullanılan hafıza miktarını azaltırken, büyük seçilmesi ise işlem yükü artacak, en iyi sınıflandırma doğruluğuna erişim süresi uzayacak bunun yanında sınıflandırma doğruluğu artacaktır. Bu nedenle, en uygun epok sayısını 200 olarak belirlenmiştir.

Uygun ağ yapısını belirlemek amacıyla 30 parçacığa sahip ağın gizli katmanında bulunan hücre sayısında değişiklikler yapılarak 200 epok ile eğitim ve test işlemleri gerçekleştirilmiş, sınıflandırma doğrulukları ve duyarlılık - belirlilik analizleri incelenmiştir. Başlangıçta, gizli katmanda 3 hücre kullanılması ile 19 (5x3+4x1) boyutlu bir parçacık sürüsü elde edilmiştir. Burada 5, gizli katmana gelen toplam bağlantı sayısı (eşik dahil), 3 gizli katman hücre sayısı, 4 çıkış katmanına gelen toplam bağlantı sayısı (eşik dahil), 1 çıkış katmanı hücre sayısıdır.

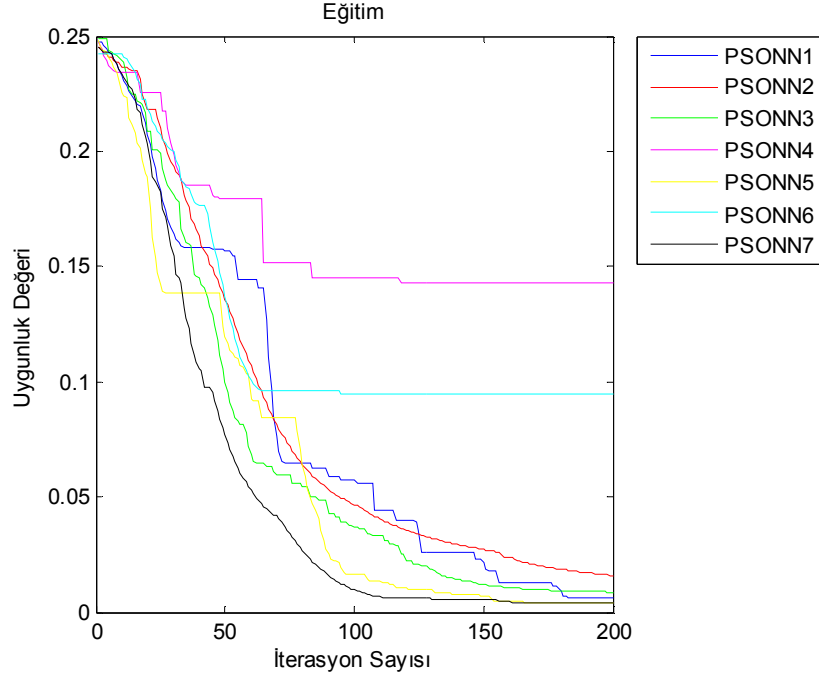
Hücre sayısının 3 seçilmesi ile elde edilen ağların, 200 epokta gbest değerleri ve doğruluk oranları Çizelge 4.13.' te, duyarlılık ve belirlilik analizi sonuçları Çizelge 4.14.' te ve iterasyon sayısına göre uygunluk değerinin değişimini gösteren grafik ise Şekil 4.9.' da verilmiştir.

Çizelge 4.13. 200 epokta 3 hücre için ağların performansı

Ağ Tipi	Gbest	% Eğitim Doğruluk Oranı	% Test Doğruluk Oranı
PSOINN1	0,0066	63	99,7500
PSOINN2	0,0159	90,5833	99,2500
PSOINN3	0,0087	99	99,2500
PSOINN4	0,1427	75,5000	89,2500
PSOINN5	0,0037	50	99,5000
PSOINN6	0,0947	50	88,7500
PSOINN7	0,0039	98,2500	100

Çizelge 4.14. 200 epokta 3 hücre için duyarlılık-belirlilik değerleri

Ağ Tipi	Eğitim		Test	
	Duyarlılık	Belirlilik	Duyarlılık	Belirlilik
PSOINN1	0,2600	1	0,9900	1
PSOINN2	0,8117	1	0,9850	1
PSOINN3	0,9883	0,9917	0,9850	1
PSOINN4	0,6533	0,8567	0,8850	0,9000
PSOINN5	1	0	1	0,9900
PSOINN6	1	0	0,7750	1
PSOINN7	1	0,9650	1	1



Şekil 4.9. 200 epokta 3 hücre için iterasyon sayısı – uygunluk değeri değişim grafiği

Gizli katman hücre sayısının 4 olarak belirlenmesi sonucunda elde edilen doğruluk oranları, duyarlılık ve belirlilik analizi sonuçları ve iterasyon sayısı – uygunluk değeri değişim grafiği sırasıyla Çizelge 4.15., Çizelge 4.16. ve Şekil 4.10.’ da verilmiştir. Ayrıca gizli katmanda 4 hücre kullanılması ile 25 (5x4+5x1) boyutlu bir parçacık sürüsü elde edilmiştir.

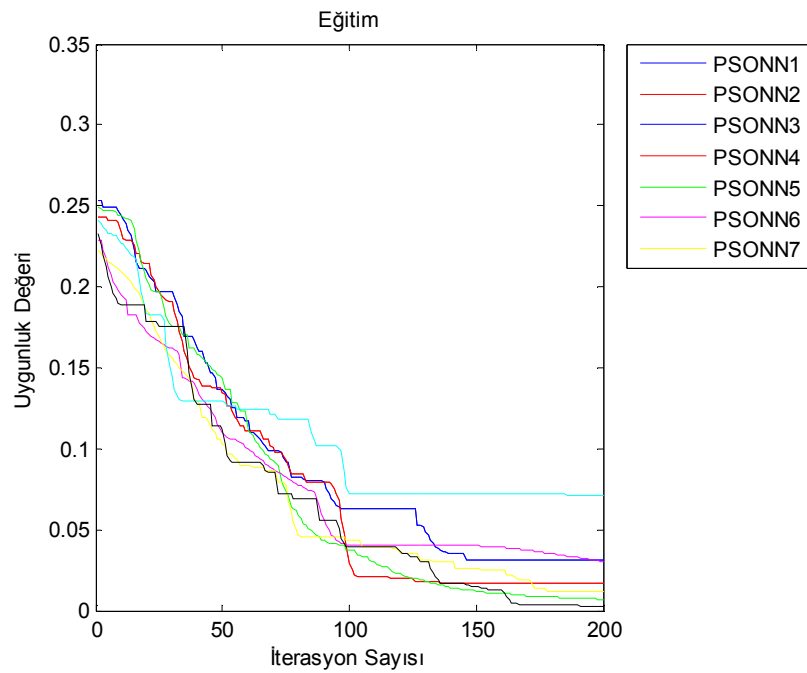
PSONN7 dışında diğer ağlar istenilen sonuçları verememiştir.

Çizelge 4.15. 200 epokta 4 hücre için ağların performansı

Ağ Tipi	Gbest	% Eğitim Doğruluk Oranı	% Test Doğruluk Oranı
PSONN1	0,0308	50	97,7500
PSONN2	0,0172	50	99
PSONN3	0,0070	92,6667	99,2500
PSONN4	0,0304	69,8333	96,2500
PSONN5	0,0114	84,6667	99
PSONN6	0,0707	72,7500	93,2500
PSONN7	0,0029	99,5833	100

Çizelge 4.16. 200 epokta 4 hücre için duyarlılık-belirlilik değerleri

Ağ Tipi	Eğitim		Test	
	Duyarlılık	Belirlilik	Duyarlılık	Belirlilik
PSONN1	1	0	1	0,9550
PSONN2	1	0	0,9800	1
PSONN3	0,8533	1	0,9850	1
PSONN4	0,3967	1	0,9250	1
PSONN5	1	0,6933	1	0,9800
PSONN6	0,4550	1	0,8650	1
PSONN7	0,995	0,9967	1	1



Şekil 4.10. 200 epokta 4 hücre için iterasyon sayısı – uygunluk değeri değişim grafiği

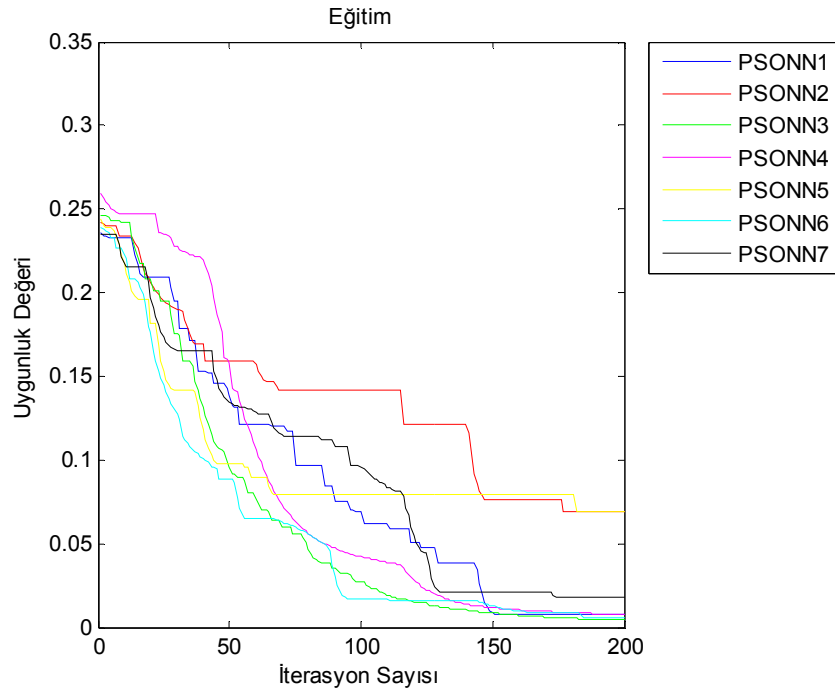
Gizli katman hücre sayısının 5 olarak belirlenmesi sonucunda elde edilen doğruluk oranları, duyarlılık ve belirlilik analizi sonuçları ve iterasyon sayısı – uygunluk değeri değişim grafiği sırasıyla Çizelge 4.17., Çizelge 4.18. ve Şekil 4.11.’ de verilmiştir. PSONN1 ve PSONN3 iyi sınıflandırma gerçekleştirmişlerdir. Gizli katmanda 5 hücre kullanılması ile 31 ($5 \times 5 + 6 \times 1$) boyutlu bir parçacık sürüsü elde edilmiştir. Görüldüğü üzere hücre sayısının artması ile parçacık sürüsünün boyutu ve işlem yükü artmakta, dolayısı ile hedefe ulaşma süresi azalmaktadır.

Çizelge 4.17. 200 epokta 5 hücre için ağların performansı

Ağ Tipi	Gbest	% Eğitim Doğruluk Oranı	% Test Doğruluk Oranı
PSONN1	0,0080	96,4167	98,7500
PSONN2	0,0689	50	93,5000
PSONN3	0,0043	98,1667	99,7500
PSONN4	0,0074	91,0833	99,2500
PSONN5	0,0692	68,7500	93
PSONN6	0,0053	79,9167	100
PSONN7	0,0177	50	98,2500

Çizelge 4.18. 200 epokta 5 hücre için duyarlılık-belirlilik değerleri

Ağ Tipi	Eğitim		Test	
	Duyarlılık	Belirlilik	Duyarlılık	Belirlilik
PSONN1	0,9283	1	0,9750	1
PSONN2	1	0	1	0,8700
PSONN3	0,9633	1	0,9950	1
PSONN4	1	0,8217	0,9850	1
PSONN5	0,3750	1	0,8600	1
PSONN6	0,5983	1	1	1
PSONN7	1	0	0,9650	1



Şekil 4.11. 200 epokta 5 hücre için iterasyon sayısı – uygunluk değeri değişim grafiği

Hücre sayısının farklı seçilmesi durumunda ağlar, birbirleri arasında herhangi bir ilişki bulunmayan sonuçlar vermiştir. Örneğin, Çizelge 4.13.' te görüldüğü gibi PSONN2' in eğitim doğruluğu %90,58 iken Çizelge 4.15.' te %50 ve Çizelge 4.17.' de %50 olmuştur. PSONN5 için ise sırası ile %50, %84,67 ve %68,75 değerleri elde edilmiştir. Hız kazanmak amacı ile de en uygun hücre sayısı 3 olarak belirlenmiştir.

Ayrıca, parçacık sayıları değiştirilerek elde edilen sonuçlara bakılarak en uygun parçacık sayısı belirlenmeye çalışılmıştır. Parçacık sayıları, genellikle 20 ile 40 arasındadır (Tamer ve Karakuzu, 2006). Bu nedenle, parçacık sayıları; 20, 30 ve 40 seçilmiştir. Ağ yapısı {4, 3, 1} olarak belirlenmiştir, burada, ilk değer olan (4); giriş katmanındaki hücre sayısını, ikinci değer (3), gizli katmandaki hücre sayısını, son değer (1) ise, teşhis sonucunu belirtmektedir. Epok sayısı ise 200 seçilmiştir.

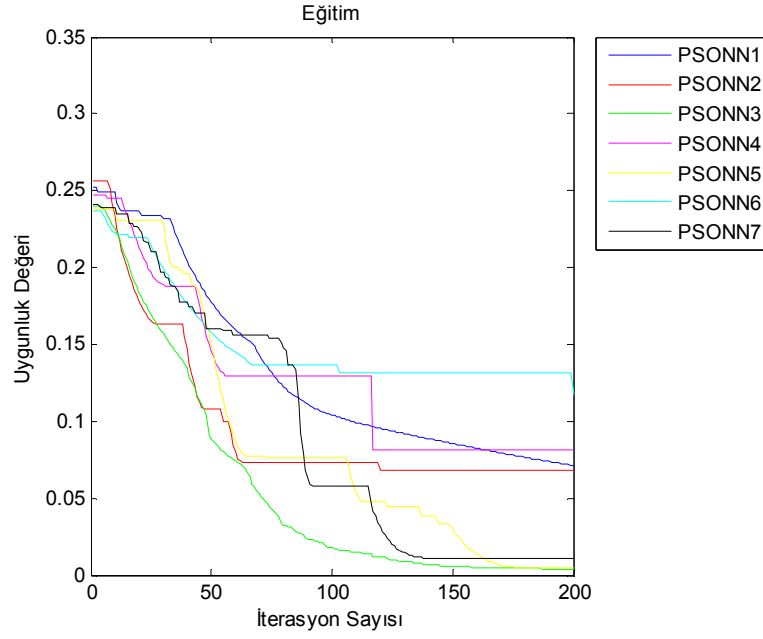
20 parçacık kullanılması sonucunda elde edilen doğruluk oranları Çizelge 4.19.' ta, duyarlılık ve belirlilik analizi sonuçları Çizelge 4.20.' de ve iterasyon sayısı – uygunluk değeri değişim grafiği Şekil 4.12.' de verilmiştir. En iyi sınıflandırma oranlarını veren ağ, PSONN3' tür.

Çizelge 4.19. 200 epokta 20 parçacık için ağların performansı

Ağ Tipi	Gbest	% Eğitim Doğruluk Oranı	% Test Doğruluk Oranı
PSOENN1	0,0712	91,6667	96,2500
PSOENN2	0,0685	83,4167	92,2500
PSOENN3	0,0038	97,8333	99,5000
PSOENN4	0,0817	50	81,5000
PSOENN5	0,0046	91,2500	100
PSOENN6	0,1174	50	58
PSOENN7	0,0110	79,3333	99,2500

Çizelge 4.20. 200 epokta 20 parçacık için duyarlılık-belirlilik değerleri

Ağ Tipi	Eğitim		Test	
	Duyarlılık	Belirlilik	Duyarlılık	Belirlilik
PSOENN1	0,8333	1	0,9250	1
PSOENN2	0,6683	1	0,8450	1
PSOENN3	1	0,9567	0,9950	0,9950
PSOENN4	1	0	1	0,6300
PSOENN5	1	0,8250	1	1
PSOENN6	1	0	1	0,1600
PSOENN7	0,5867	1	0,9850	1



Şekil 4.12. 200 epokta 20 parçacık için iterasyon sayısı – uygunluk değeri değişim grafiği

30 parçacık kullanılması sonucunda elde edilen doğruluk oranları, duyarlılık ve belirlilik analizi sonuçları ve iterasyon sayısı – uygunluk değeri değişim grafiği sırasıyla Çizelge 4.21., Çizelge 4.22. ve Şekil 4.13.’ te verilmiştir.

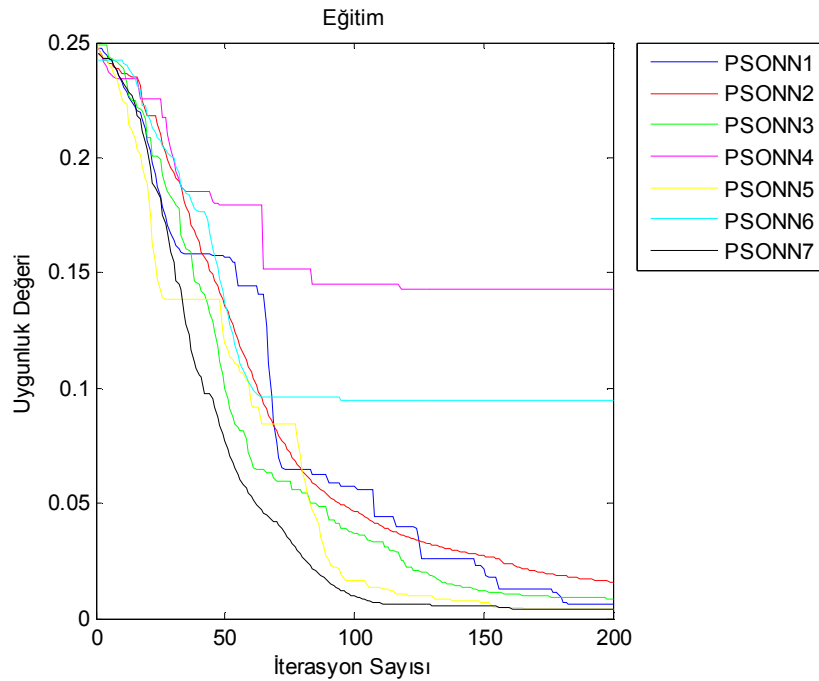
En iyi sınıflandırma doğruluklarını PSONN7’ nin verdiği görülmüştür.

Çizelge 4.21. 200 epokta 30 parçacık için ağların performansı

Ağ Tipi	Gbest	% Eğitim Doğruluk Oranı	% Test Doğruluk Oranı
PSONN1	0,0066	63	99,7500
PSONN2	0,0159	90,5833	99,2500
PSONN3	0,0087	99	99,2500
PSONN4	0,1427	75,5000	89,2500
PSONN5	0,0037	50	99,5000
PSONN6	0,0947	50	88,7500
PSONN7	0,0039	98,2500	100

Çizelge 4.22. 200 epokta 30 parçacık için duyarlılık-belirlilik değerleri

Ağ Tipi	Eğitim		Test	
	Duyarlılık	Belirlilik	Duyarlılık	Belirlilik
PSONN1	0,2600	1	0,9900	1
PSONN2	0,8117	1	0,9850	1
PSONN3	0,9883	0,9917	0,9850	1
PSONN4	0,6533	0,8567	0,8850	0,9000
PSONN5	1	0	1	0,9900
PSONN6	1	0	0,7750	1
PSONN7	1	0,9650	1	1



Şekil 4.13. 200 epokta 30 parçacık için iterasyon sayısı – uygunluk değeri değişim grafiği

40 parçacık kullanılması sonucunda elde edilen doğruluk oranları, duyarlılık ve belirlilik analizi sonuçları ve iterasyon sayısı – uygunluk değeri değişim grafiği sırasıyla Çizelge 4.23., Çizelge 4.24. ve Şekil 4.14.' te verilmiştir.

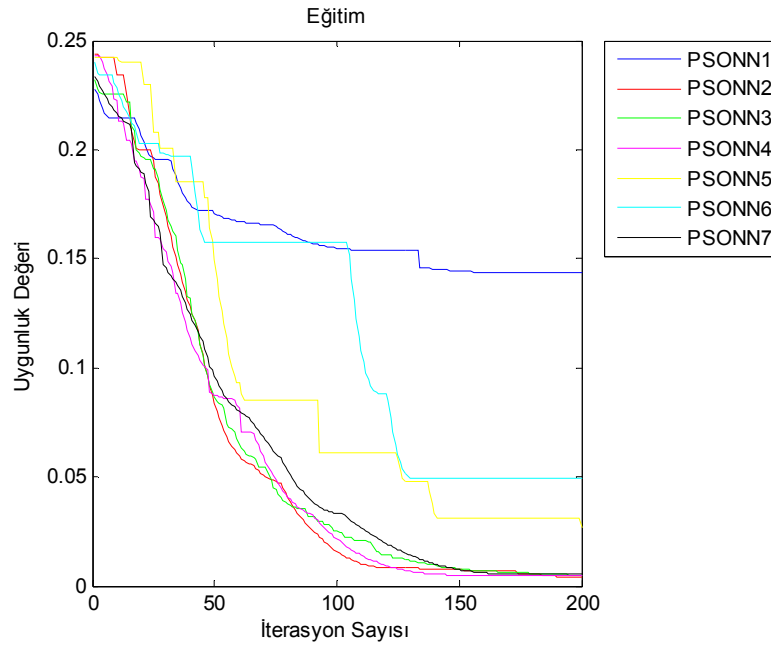
Her bir ağın eğitiminde 20, 30 ve 40 adet parçacık kullanılması ile sonuç olarak elde edilen doğruluk oranlarının, parçacık sayısı ile doğru ya da ters orantılı olup olmadığı hakkında bir şey söylenemez. Çünkü 20 parçacıkta en iyi sınıflandırmayı gerçekleştiren PSONN3 iken 30 parçacıkta PSONN7, 40 parçacıkta ise eğitim kısmında en iyi sınıflandırmayı PSONN3 test kısmında ise PSONN2 gerçekleştirmiştir.

Çizelge 4.23. 200 epokta 40 parçacık için ağların performansı

Ağ Tipi	Gbest	% Eğitim Doğruluk Oranı	% Test Doğruluk Oranı
PSONN1	0,1434	50	95,7500
PSONN2	0,0042	99,3333	100
PSONN3	0,0050	99,4167	99,7500
PSONN4	0,0045	97,5000	99,5000
PSONN5	0,0267	50	97,2500
PSONN6	0,0491	59,5833	95,5000
PSONN7	0,0054	61,0833	99,2500

Çizelge 4.24. 200 epokta 40 parçacık için duyarlılık-belirlilik değerleri

Ağ Tipi	Eğitim		Test	
	Duyarlılık	Belirlilik	Duyarlılık	Belirlilik
PSONN1	1	0	0,9400	0,9750
PSONN2	0,9950	0,9917	1	1
PSONN3	0,9967	0,9917	0,9950	1
PSONN4	1	0,9500	0,9900	1
PSONN5	1	0	0,9450	1
PSONN6	0,1917	1	0,9100	1
PSONN7	1	0,2217	0,9950	0,9900



Şekil 4.14. 200 epokta 40 parçacık için iterasyon sayısı – uygunluk değeri değişim grafiği

Çizelge 4.19., Çizelge 4.21. ve Çizelge 4.23.' te görüldüğü gibi ortalama en iyi sınıflandırma doğrulukları 40 parçacık için elde edilmiştir. Hema ve ark. (2008) çalışmalarında, ağ eğitimlerinde bu çalışmada kullanılan veri miktarından daha az sayıdaki verinin eğitimi için 25 parçacık kullanmışlardır. Ancak daha düşük parçacık seçilmesi ile hesaplama karmaşıklığını azaltacaktır. Bu sebeplerle, 30 ve 40 parçacık için ağlar ayrı ayrı 10 kez koşturulmuş, en uygun parçacık sayısı belirlenmeye çalışılmıştır.

Epok sayısı 200, parçacık sayısı 30 ve gizli katman hücre sayısı 3 seçilmiş, ağlar 10 kez koşturulmuş ve Çizelge 4.25.' teki sonuçlar elde edilmiştir. Çizelge 4.25' de görüldüğü gibi en iyi sınıflandırmayı gerçekleştiren ağlar;

- PSONN3: Eğitim kısmında %99,67, test kısmında ise %100 oranında başarımlı sağlamıştır.
- PSONN7: Eğitim kısmında %98,25, test kısmında ise %99,5 oranında başarımlı sağlamıştır.

Eğitim ve test kısımlarında sırasıyla %94,5 ve %100 sınıflandırma doğruluğu verdiği için PSONN2 de en iyiye yakındır.

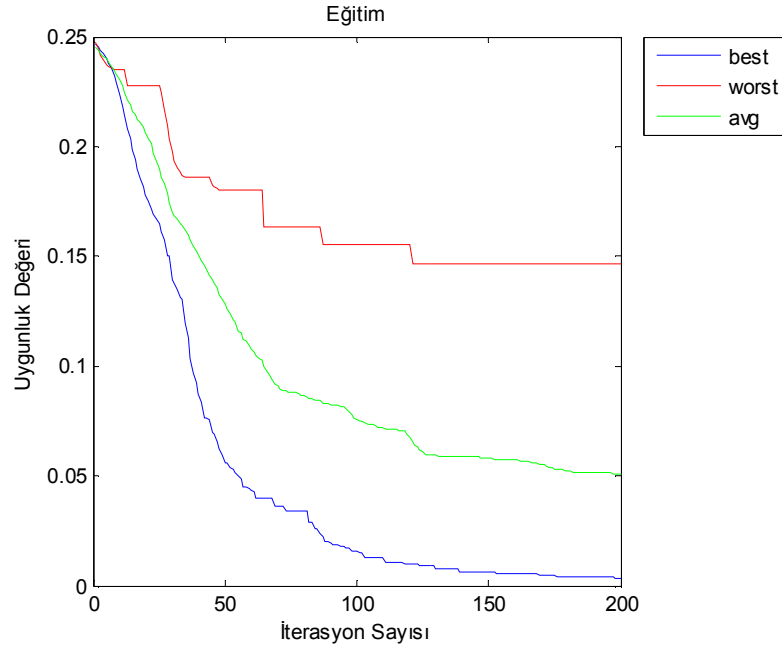
En kötü sınıflandırmayı gerçekleştiren ağın, eğitim kısmında %50 oranını vererek PSONN6 olduğu görülmüştür.

Çizelge 4.25. 30 parçacık için ağların 200 epokla 10 kez koşturulması ile elde edilen değerler

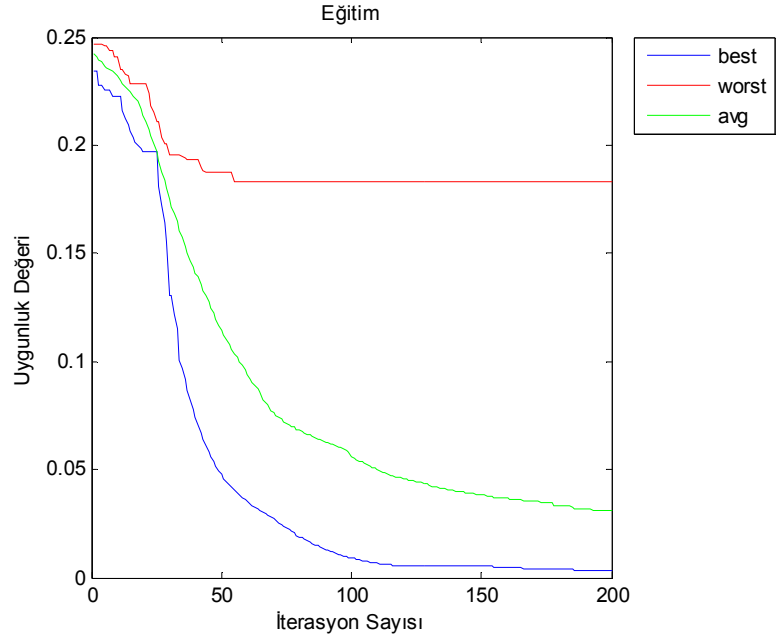
Ağ Tipi	Best	Worst	% Eğitim Doğruluk Oranı	% Test Doğruluk Oranı
PSOENN1	0,0034	0,1463	87,1667	100
PSOENN2	0,0031	0,1839	94,5000	100
PSOENN3	0,0041	0,1701	99,6667	100
PSOENN4	0,0044	0,1684	80,3333	100
PSOENN5	0,0034	0,1463	87,1667	100
PSOENN6	0,0104	0,0973	50	99,5000
PSOENN7	0,0061	0,1784	98,2500	99,5000

PSOENN1, PSOENN2, PSOENN3, PSOENN4, PSOENN5, PSOENN6 ve PSOENN7 ağlarının eğitimi sırasında elde edilen en iyi (best), en kötü (worst) ve ortalama (avg) uygunluk değerlerini veren işlem süreçleri, sırasıyla Şekil 4.15., Şekil 4.16., Şekil 4.17., Şekil 4.18., Şekil 4.19., Şekil 4.20. ve Şekil 4.21.' de gösterilmiştir. En iyi uygunluk değeri olarak en düşük hatayı veren, en kötü uygunluk değeri olarak ta en yüksek hatayı

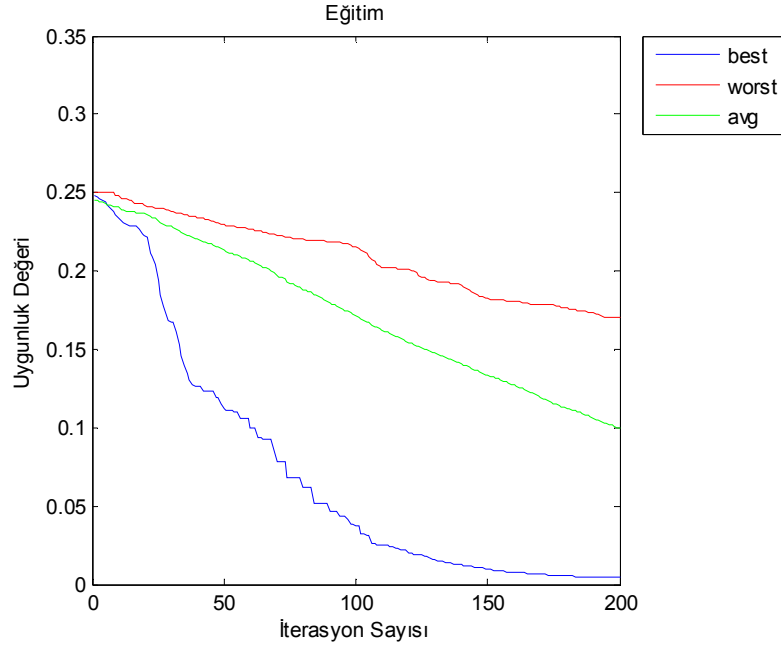
veren parçacık seçilmiş ve 200 iterasyon boyunca aldığı değerler, şekillerde gösterilmiştir.



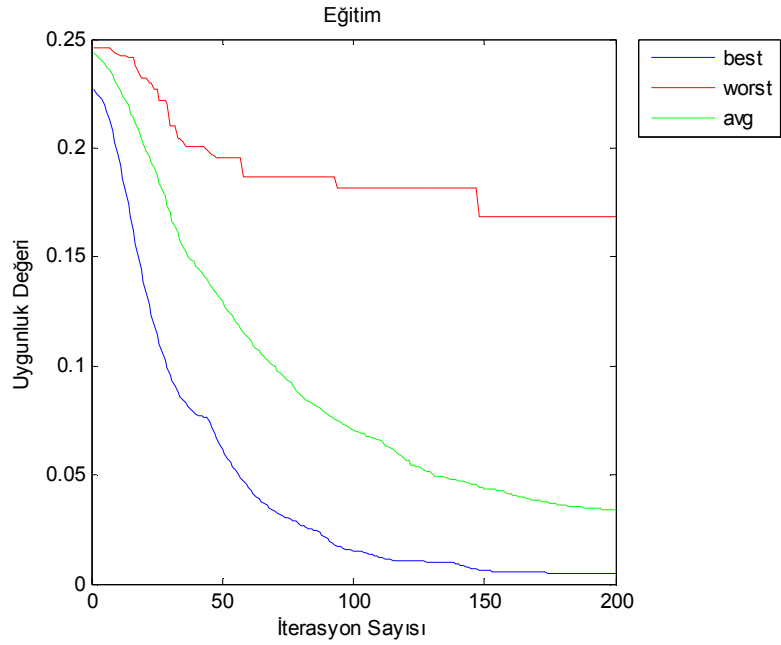
Şekil 4.15. 10 run ve 200 epokta 30 parçacık için PSNN1' in eğitim grafiği



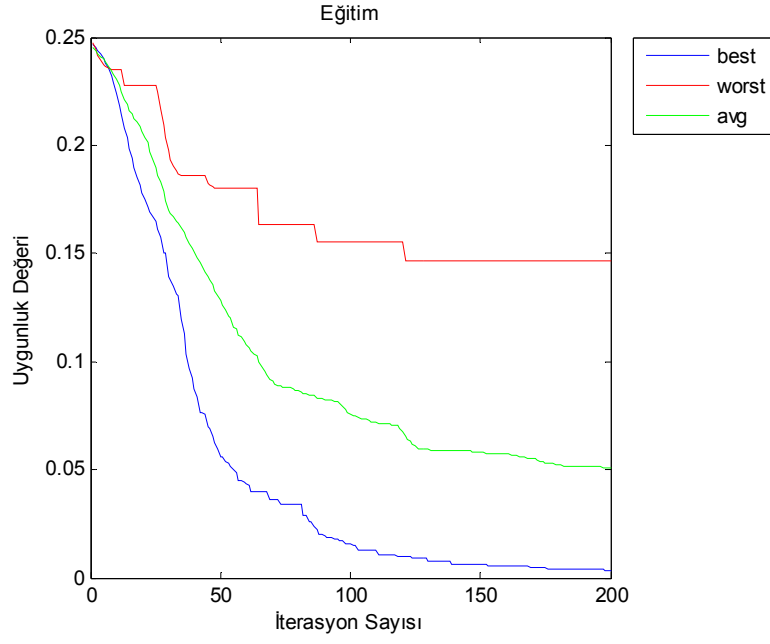
Şekil 4.16. 10 run ve 200 epokta 30 parçacık için PSNN2' nin eğitim grafiği



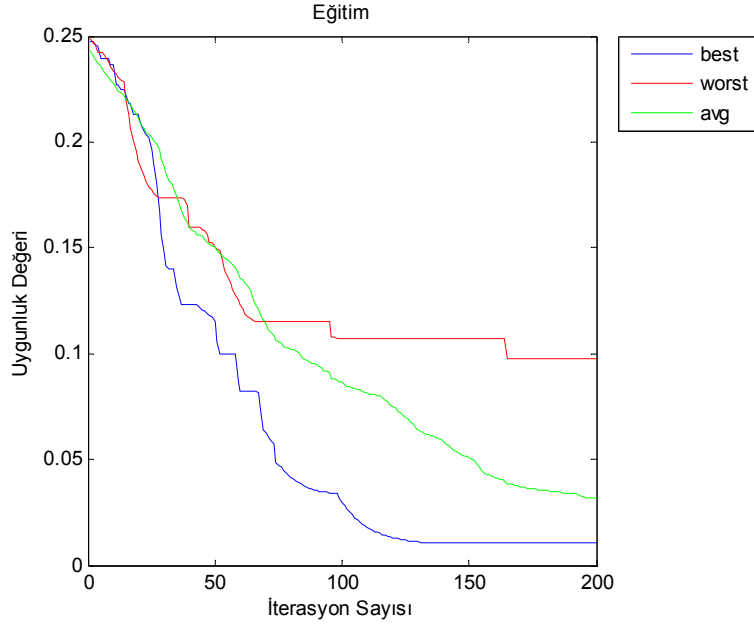
Şekil 4.17. 10 run ve 200 epokta 30 parçacık için PSNN3' ün eğitim grafiği



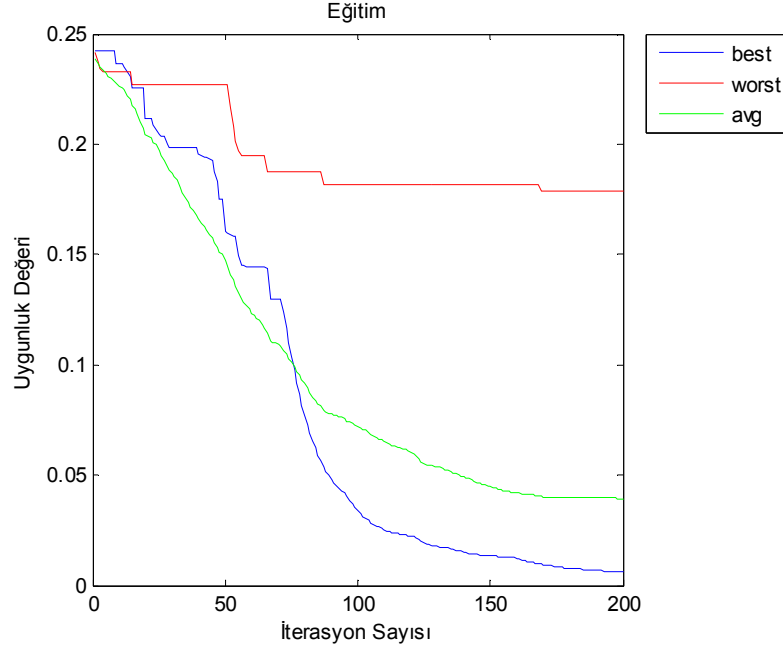
Şekil 4.18. 10 run ve 200 epokta 30 parçacık için PSNN4' ün eğitim grafiği



Şekil 4.19. 10 run ve 200 epokta 30 parçacık için PSOINN5' in eğitim grafiği



Şekil 4.20. 10 run ve 200 epokta 30 parçacık için PSOINN6' nın eğitim grafiği



Şekil 4.21. 10 run ve 200 epokta 30 parçacık için PSONN7' nin eğitim grafiği

Elde edilen en iyi uygunluk değerleri, PSONN6 ve PSONN7 ağlarınıninki hariç birbirlerine çok yakındır ve 0,30 – 0,45 arasında değişmektedir. Ortalama (avg) uygunluk değeri değişimlerinde ise en kötü ortalamaya sahip olan ağın PSONN3 olduğu görülmektedir.

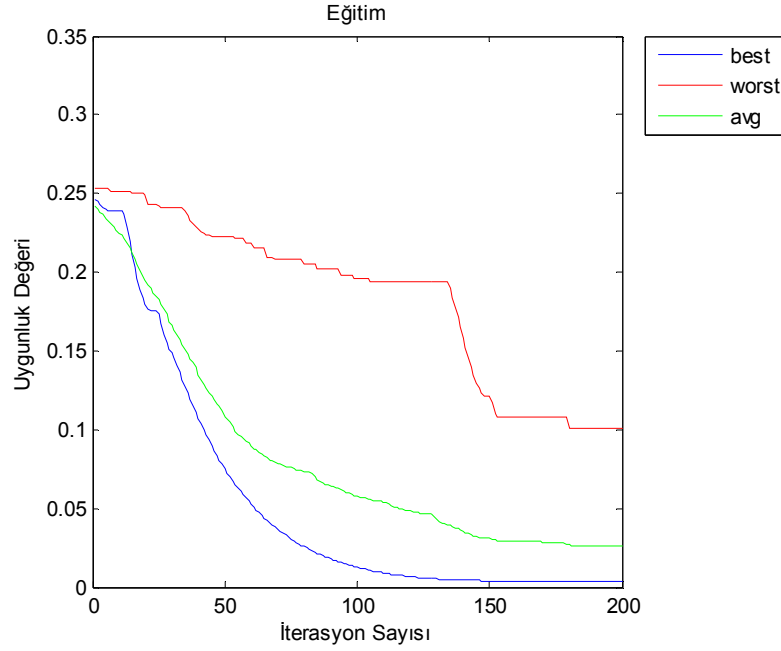
Epok sayısı 200, parçacık sayısı 40 ve gizli katman hücre sayısı 3 seçilmiş, ağlar 10 kez koşturulmuş ve Çizelge 4.26.' daki sonuçlar elde edilmiştir.

Çizelge 4.26. 40 parçacık için ağların 200 epokla 10 kez koşturulması ile elde edilen değerler

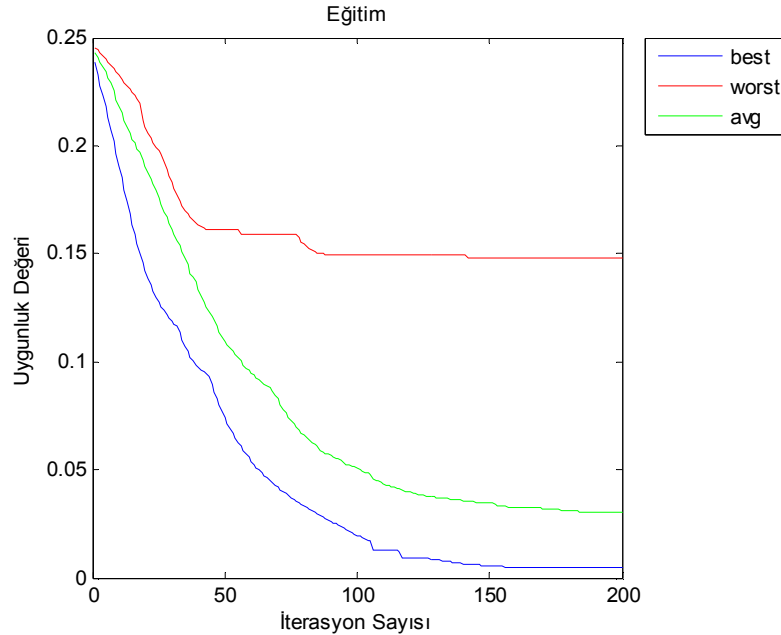
Ağ Tipi	Best	Worst	% Eğitim Doğruluk Oranı	% Test Doğruluk Oranı
PSONN1	0,0040	0,1012	85,5833	99,7500
PSONN2	0,0045	0,1482	50	100
PSONN3	0,0386	0,1987	94	99
PSONN4	0,0016	0,0399	82,7500	100
PSONN5	0,0042	0,1417	69,4167	99,5000
PSONN6	0,0048	0,1785	64,5000	100
PSONN7	0,0060	0,1602	50	99

Çizelge 4.26' da görüldüğü gibi en iyi sınıflandırmayı gerçekleştiren PSONN3 ağının eğitim doğruluk oranı %94, test doğruluk oranı ise %99 oranında başarımlı sağlamıştır. En kötü sınıflandırmayı gerçekleştiren ağların, eğitim kısmında %50 doğruluk oranını vererek PSONN2 ve PSONN7 olduğu görülmüştür.

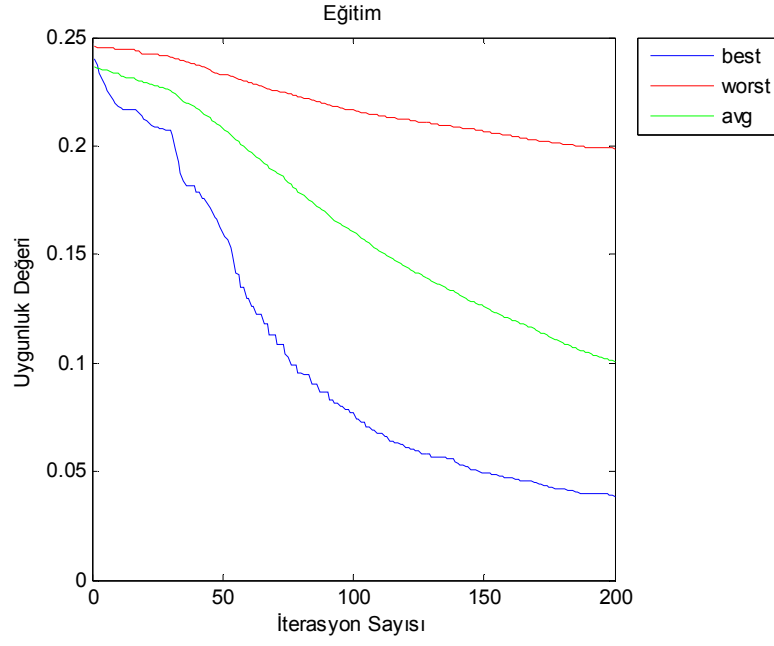
40 parçacık ile eğitilen PSONN ağlarının eğitimi sırasında elde edilen en iyi, en kötü ve ortalama uygunluk değerlerini veren işlem süreçleri, sırasıyla Şekil 4.22., Şekil 4.23., Şekil 4.24., Şekil 4.25., Şekil 4.26., Şekil 4.27. ve Şekil 4.28.' de gösterilmiştir.



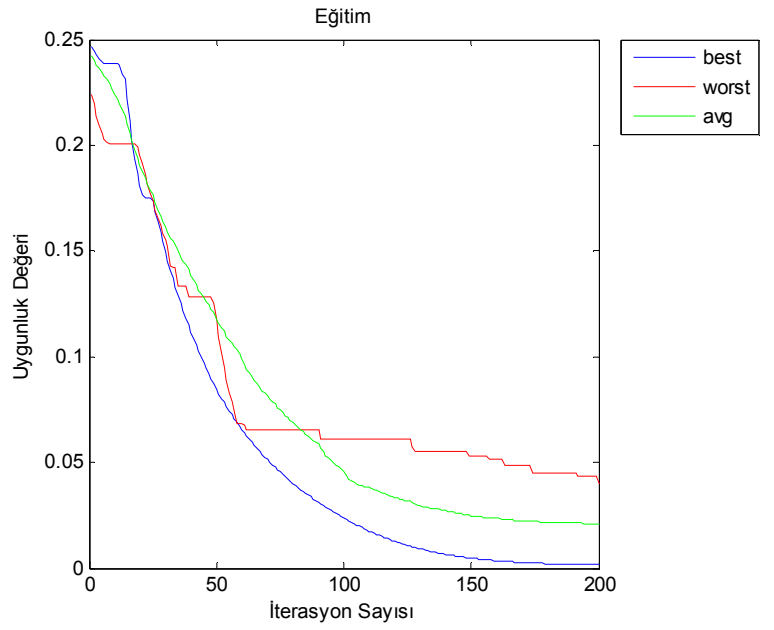
Şekil 4.22. 10 run ve 200 epokta 40 parçacık için PSONN1' in eğitim grafiği



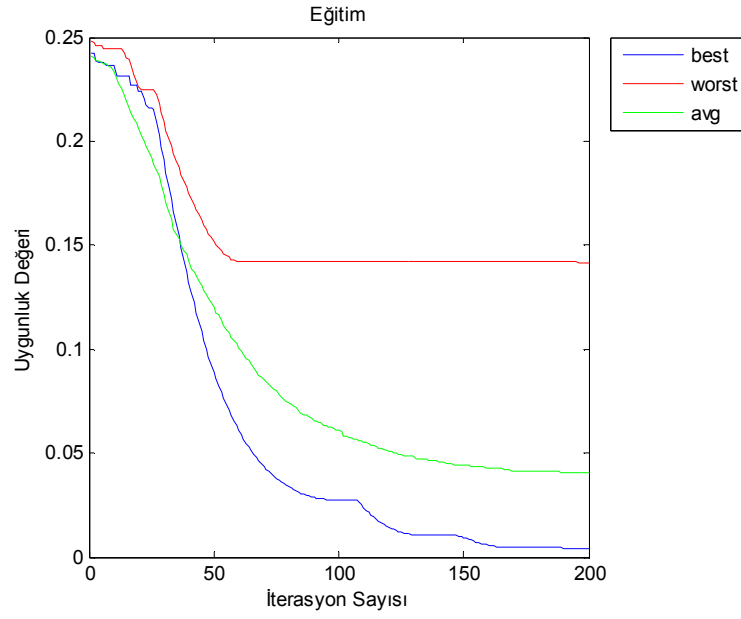
Şekil 4.23. 10 run ve 200 epokta 40 parçacık için PSONN2' nin eğitim grafiği



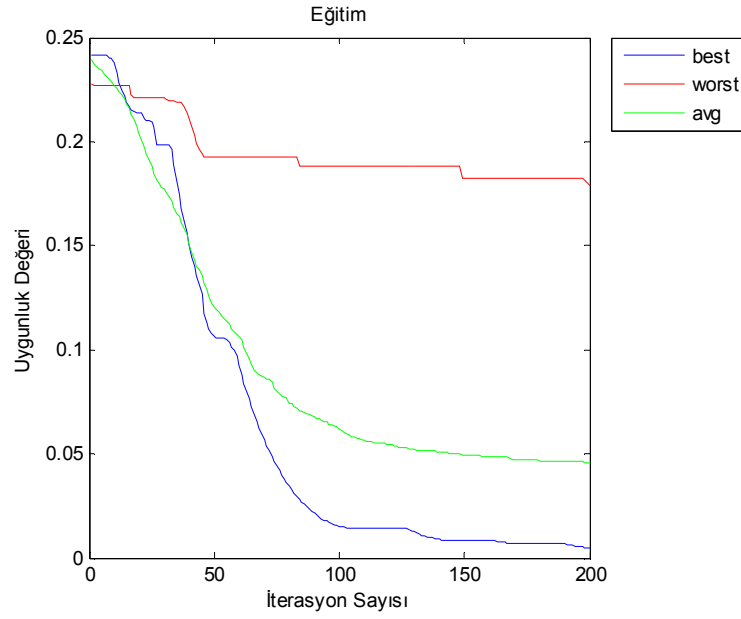
Şekil 4.24. 10 run ve 200 epokta 40 parçacık için PSNN3' ün eğitim grafiği



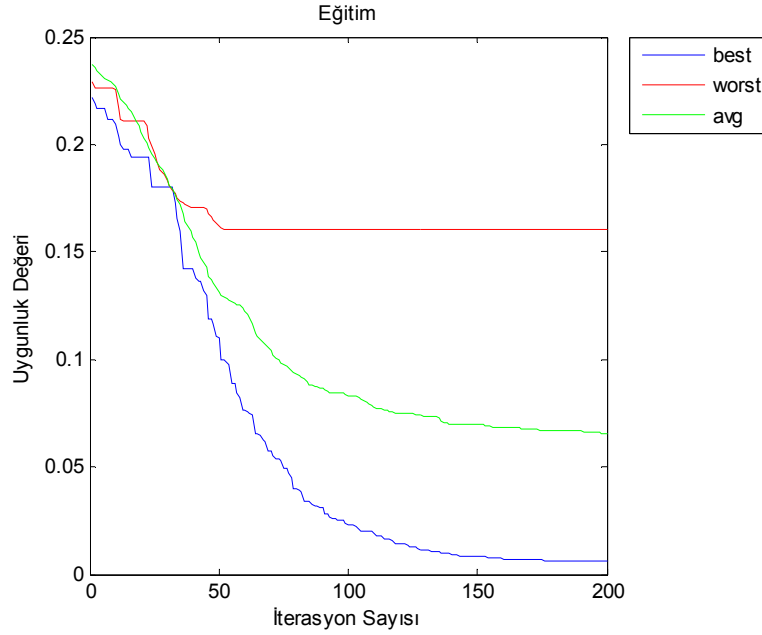
Şekil 4.25. 10 run ve 200 epokta 40 parçacık için PSNN4' ün eğitim grafiği



Şekil 4.26. 10 run ve 200 epokta 40 parçacık için PSONN5' in eğitim grafiği



Şekil 4.27. 10 run ve 200 epokta 40 parçacık için PSONN6' nın eğitim grafiği



Şekil 4.28. 10 run ve 200 epokta 40 parçacık için PSONN7' nin eğitim grafiği

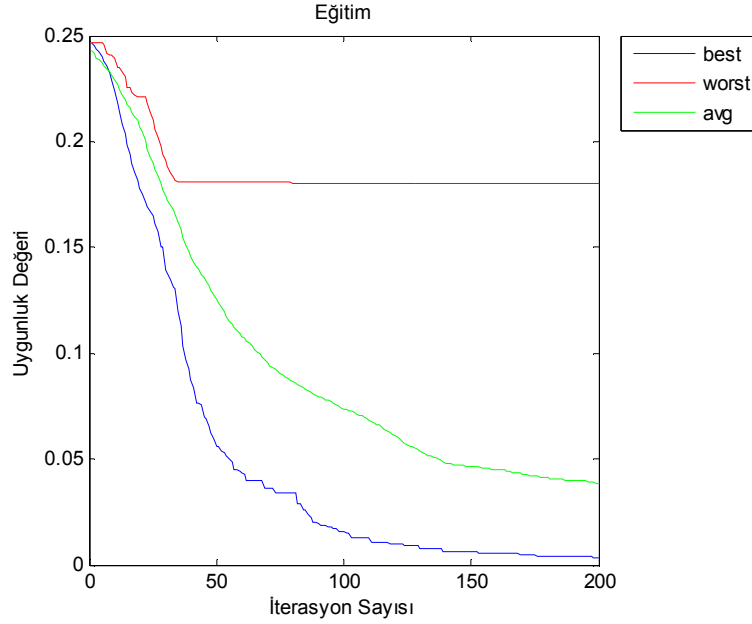
Çizelge 4.25. ve 4.26.' deki eğitim ve test doğruluk oranları karşılaştırıldığında, 30 parçacık kullanan ağların, 40 parçacık kullanan ağlara oranla daha iyi sınıflandırma gerçekleştirdiği görülmektedir.

PSO yöntemleri içerisinde en iyi olanı belirlemek amacıyla en uygun gizli katman hücre sayısına (3) sahip ağlar, en uygun parçacık (30) ve epok sayısı (200) kullanılarak 30 kez koşturulmuş ve Çizelge 4.27.' deki değerler elde edilmiştir. Çizelge 4.27.' de görüldüğü gibi, test işleminde PSONN6 ağı hariç %100 doğru sınıflandırma gerçekleştirilmiştir. Eğitim kısmında en iyi sınıflandırma işlemini, eğitim doğruluk oranı %98 üzerinde olan PSONN3, PSONN4 ve PSONN2 gerçekleştirmiştir.

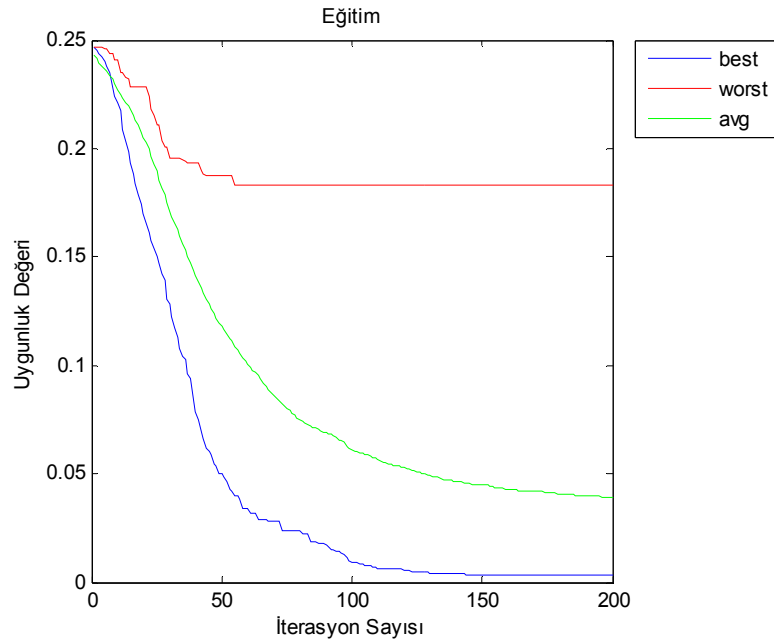
Çizelge 4.27. Ağların 30 kez koşturulması ile elde edilen performans değerleri

Ağ Tipi	Best	Worst	% Eğitim Doğruluk Oranı	% Test Doğruluk Oranı
PSONN1	0,0034	0,1799	87,1667	100
PSONN2	0,0031	0,1829	98,1667	100
PSONN3	0,0041	0,2289	99,6667	100
PSONN4	0,0038	0,1677	98,7500	100
PSONN5	0,0034	0,1799	87,1667	100
PSONN6	0,0047	0,2125	84,5000	99,2500
PSONN7	0,0059	0,1784	79,0833	100

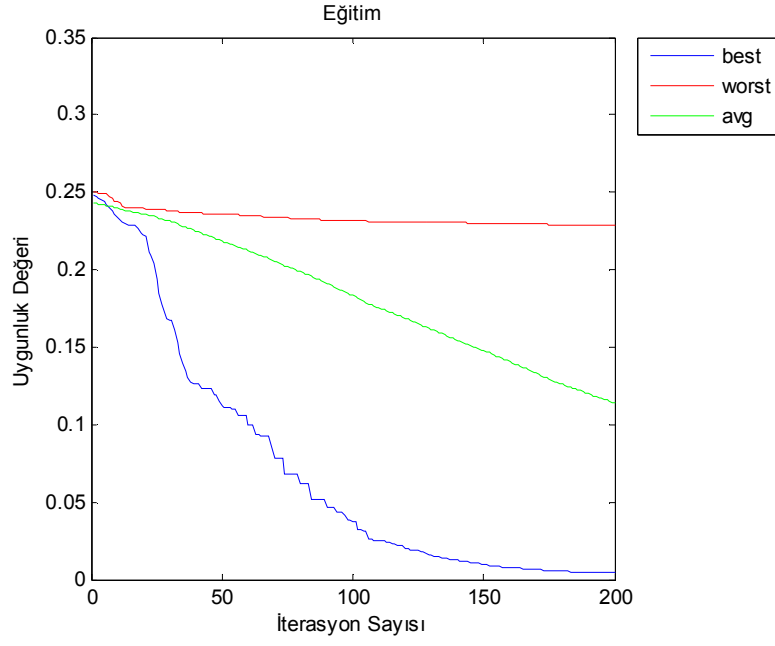
Gizli katmanında 3 hücre bulunan PSONN ağlarının, 30 parçacık ve 200 epok ile 30 kez koşturulması sonucunda elde edilen en iyi, en kötü ve ortalama uygunluk değerlerini veren işlem süreçleri, sırasıyla Şekil 4.29., Şekil 4.30., Şekil 4.31., Şekil 4.32., Şekil 4.33., Şekil 4.34. ve Şekil 4.35.' te gösterilmiştir.



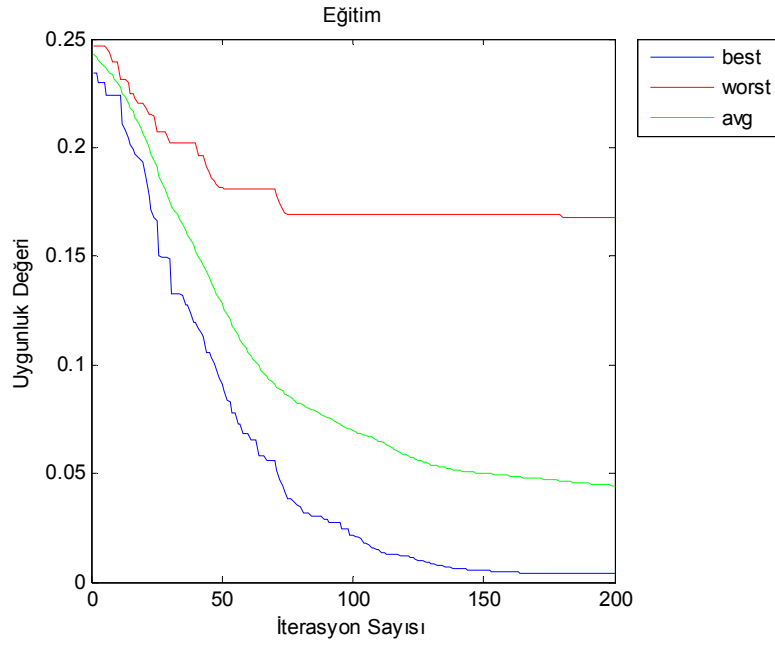
Şekil 4.29. 30 run sonucu PSONN1' in eğitim grafiği



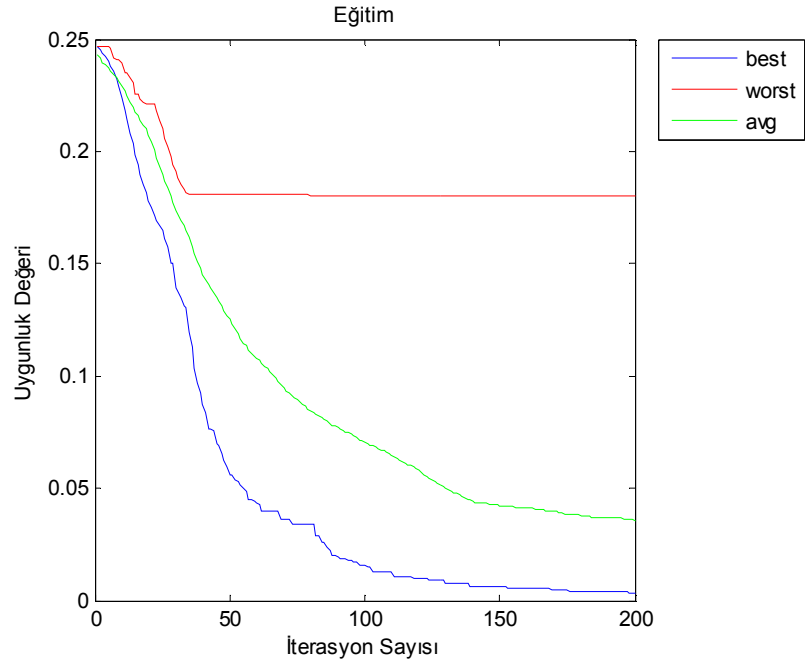
Şekil 4.30. 30 run sonucu PSONN2' nin eğitim grafiği



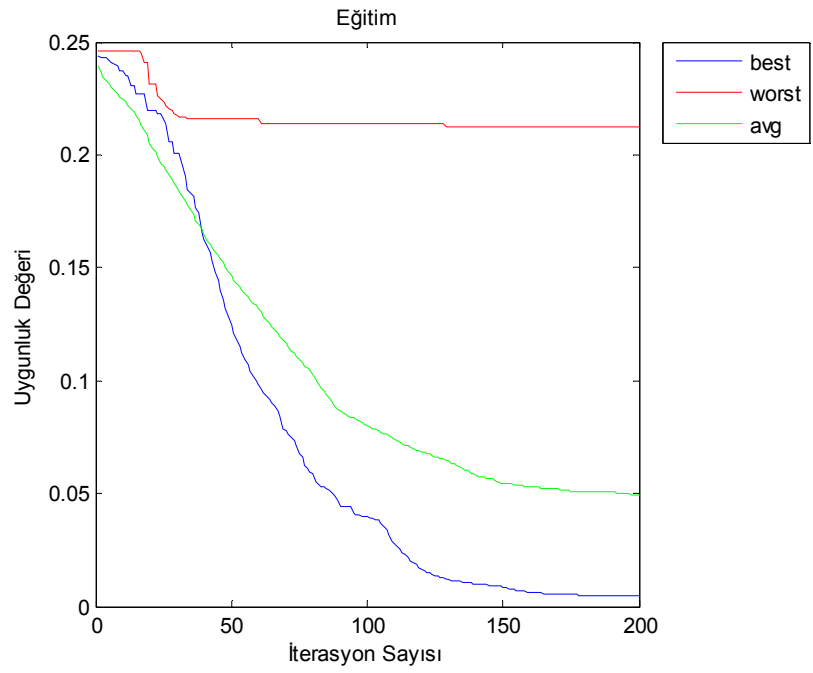
Şekil 4.31. 30 run sonucu PSNN3' ün eğitim grafiği



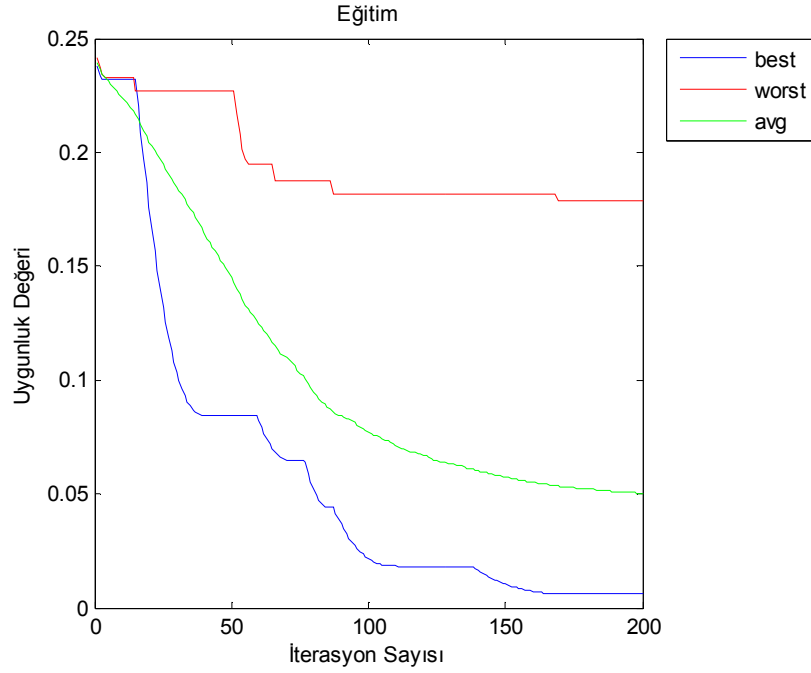
Şekil 4.32. 30 run sonucu PSNN4' ün eğitim grafiği



Şekil 4.33. 30 run sonucu PSONN5' in eğitim grafiği



Şekil 4.34. 30 run sonucu PSONN6' nın eğitim grafiği



Şekil 4.35. 30 run sonucu PSONN7' nin eğitim grafiği

Şekillerde görüldüğü gibi, en iyi, en kötü ve ortalama uygunluk değerleri yaklaşık olarak 0,25 seviyelerinde başlamaktadır.

PSONN ağlarının 30 kez koşturulması ile elde edilen uygunluk değerleri içerisinde en iyi olanların 0,0030 ile 0,0060 arasında, en kötü olanların ise 0,16 ile 0,23 arasında değiştiği görülmektedir.

Ağların hız hesaplarında, PSO'yu en temel hali ile kullanan PSONN1 hariç diğerlerinde eylemsizlik ağırlığı, sınırlama katsayısının hesaplanması gibi ek matematiksel işlemler yapılmıştır. Bu değerlerin kullanılması, hedefe yakınsamayı sağlayarak sınıflandırma başarımını artırmıştır.

5. SONUÇLAR VE ÖNERİLER

5.1. Sonuçlar

Bu tez çalışmasında, epilepsi hastalarının teşhisi amacıyla yapay sinir ağlarının kullanımına ilişkin sonuçlar sunulmuştur. Yapay sinir ağlarının eğitilmesinde PSO çeşitleri ve geri yayılım algoritması kullanılmıştır. YSA programları, MATLAB® ile yazılmıştır. Programlar değişik hücre sayıları (3, 4 ve 5), parçacık sayıları (20, 30, 40) ve epok sayıları (50,100, 150, 200 ve 250) verilerek sınanmıştır. Yapılan denemeler sonucunda, en uygun hücre sayısı 3, parçacık sayısı 30 ve epok sayısı 200 olarak belirlenmiştir.

İşlem yükü ve gerekli hafıza miktarının, parçacık sayısına ve ağı katmanlarındaki hücre sayısına bağlı olarak doğrusal oranda değiştiği belirlenmiştir. Parçacık sayısı arttıkça başarı artmakla birlikte ağı hızı çok yavaşlamaktadır.

Bu çalışmada, Çizelge 4.25.' de verilen eğitim ve test verilerindeki sınıflandırma doğruluk oranlarına bakıldığında en başarılı YSA yapılarının PSONN2, PSONN3 ve PSONN4 olduğu söylenebilir. Bu ağların eğitim başarı yüzdeleri %98,17, %99,67 ve %98,75' tir. Eğitmenin ardından yapılan test sonucunda elde edilen değerlere bakıldığında ise en düşük test başarısının %99,25 ve en yüksek test başarısının %100 olduğu görülmektedir.

Ayrıca, benzer bir çalışma olan Hema ve ark. (2008)' nin yaptıkları çalışmada, EEG sinyallerinin sınıflandırılması için yapay sinir ağlarından yararlanılmış, öğrenme algoritması olarak PSO kullanılmıştır. Durdurma kriteri olarak maksimum iterasyon sayısı (10.000) ulaşma veya minimum hata koşulunu (0,05) sağlamayı seçmişlerdir. Hema ve ark. (2008) yaptıkları bu çalışmada, en iyi ortalama sınıflandırma doğruluğunu %92 olarak elde etmişlerdir. Bu tez çalışmasında ise durdurma kriteri olarak iterasyon sayısı 200 ve minimum hata (MSE) koşulu 0.001 seçilmiştir. Hema ve ark. (2008) yaptıkları çalışmada elde ettikleri doğruluk oranları ile karşılaştırıldığında bu çalışmada daha iyi sonuçlar elde edildiği görülmüştür. Hema ve ark. (2008) çalışmalarına kıyasla bu çalışmada, daha az iterasyon, giriş ve gizli katman hücresi (doğal olarak daha küçük boyutta parçacık sürüsü) kullanılmıştır.

Öğrenme algoritması olarak geri yayılım algoritmasını kullanan BPNN için eğitim başarı yüzdesinin %99,83 – 99,9 olduğu, test başarı yüzdesinin ise en yüksek %58,75 bulunduğu görülmüştür. PSO kullanarak öğrenme gerçekleştiren diğer ağlar ile karşılaştırıldığında daha kötü sınıflandırma gerçekleştirmiştir.

5.2. Öneriler

Bu çalışma, temelde sınıflandırma işleminin PSO yöntemi ile optimize edilmesini amaçlamıştır. İleriki çalışmalarda PSO, YSA' nın öğrenme performansını geliştirmek için yardımcı olacaktır. Ayrıca YSA' nın başarısını arttırabilmek için herhangi bir sezgisel optimizasyon yönteminin, öğrenme algoritması olarak kullanılabileceğini göstermiştir.

Sınıflandırma işleminin otomatik bir sistem olarak tasarlanması, ilerdeki klinik kullanımlarda büyük kolaylıklar sağlama yolunu açmıştır. Bu tez çalışmasında, epilepsi hastalarının sınıflandırılmasında kullanılan yapay sinir ağı modelleri farklı tıbbi teşhis problemleri için de uyarlanabilir.

KAYNAKLAR

- Akın, M., Arserim, M. A., Kıymık, M. K. ve Türkoğlu, I., 2001, A New Approach for Diagnosing Epilepsy by Using Wavelet Transform and Neural Networks, *2001 Proceedings of the 23rd Annual EMBS International Conference*, İstanbul, 1596-1599.
- Altun, H., Eminoğlu, U. ve Tezekici, B.S, 2002, MLP Yapay Sinir Ağlarında Öğrenme Sürecinin Aktivasyon Fonksiyonu ve İstatiksel Değişim Gösteren Giriş Verilerine Bağımlılığı, *Eleco'2002 Elektrik, Elektronik ve Bilgisayar Mühendisliği Sempozyumu*, Bursa.
- Andrzejak, R.G., Lehnertz, K., Mormann, F., Rieke, C., David, P., Elger, C.E., 2001, Indications of Nonlinear Deterministic and Finite-Dimensional Structures in Time Series of brain Electrical Activity: Dependence on Recording Region and Brain State, *Phys Rev E* 2001, 64:061907.
- Avşar, E., Kamaşak, M. E. ve Çataltepe, Z., 2009, Tek-Sınıf Destek Vektör Makineleri Kullanılarak EEG İşaretlerinin Sınıflandırılması / EEG Signal Classification Using One-Class Support Vector Machines, *Biyomut 2009*, İzmir.
- Bağış, S., 2009, Yapay Zeka Algoritmaları Kullanılarak Sistem Modelleme, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü*, Kayseri.
- Bağış, A., Çetin, O., 2009, Doğrusal Olmayan Bir Sistemin Bulanık Kural Parametrelerinin PSO Algoritması İle Optimize Edilmesi, *Electronic Letters on Science & Engineering*, 5 (1).
- Barışçı, N. ve Müldür, S., 2003, Epileptik EEG Sinyallerinin Sinirsel – Bulanık Sistem ile Sınıflandırılması, *Politeknik Dergisi*, Cilt 6 (2), 445-449.
- Batar, H., 2005, EEG İşaretlerinin Dalgacık Analiz Yöntemleri Kullanılarak Yapay Sinir Ağları ile Sınıflandırılması, *Kahramanmaraş Sütçü İmam Üniversitesi Fen Bilimleri Enstitüsü*, Kahramanmaraş.
- Bezobrazova, S. ve Golovko, V., 2007, Comparative Analysis of Forecasting Neural Networks in the Application for Epilepsy Detection, *IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, 6-8 Eylül 2007, Dortmund, Germany.
- Calvalho, M. ve Ludermir, T.B., 2007, Particle Swarm Optimization of Neural Network Architectures and Weights, *Seventh International Conference on Hybrid Intelligent Systems*, Almanyada, 336-339.
- Delice, Y., 2008, Parçacık Sürü Optimizasyonu ile Yapay Sinir Ağlarından Sınıflandırma Kuralı Çıkarımı, *Erciyes Üniversitesi Sosyal Bilimler Enstitüsü*, Kayseri.
- Doğan, B., 2009, Parçacık Sürü Optimizasyonuna Dayalı Yeni Bir Aritmi Sınıflama Yöntemi, *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul.

- Der, O., Vural, R.A. ve Yıldırım, T., 2008, Parçacık Sürü Optimizasyonu Tabanlı Evirici Tasarımı (Inverter Design Based on Particle Swarm Optimization), *Yıldız Teknik Üniversitesi Elektronik ve Haberleşme Müh. Bölümü*, İstanbul.
- Eberhart, R.C. ve Shi, Y., 2001, Particle Swarm Optimization: Developments, Applications and Resources. *In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC*, Seul.
- Elmas, Ç., 2007, Yapay Zeka Uygulamaları, *Seçkin Yayıncılık*, Ankara.
- Gallad, A. E., Hawary, M. E., Sallam, A. ve Kalas, A., 2002, Enhancing the Particle Swarm Optimizer via Proper Parameter Selection, *Canadian Conference on Electrical Engineering*, 792-797.
- Gözde, H., Taplamacıoğlu, M.C., Kocaarslan, İ. ve Şenol, M.A., 2010, İki Bölge İtirmalı Termal Güç Sisteminin Yük-Frekans Kontrolü için Parçacık Sürüsü Optimizasyonu Tabanlı PI-Kontrolör, *Isı Bilimi ve Tekniđi Dergisi*, 30, 1, 13-21.
- Gu, T. Y., Ju, S.G. ve Han, F., 2009, A PSO Algorithm with the Improved Diversity for Feedforward Neural Networks, *Second International Symposium on Intelligent Information Technology and Security Informatics*, 123-127.
- Güler, İ. ve Übeyli, E. D., 2004, Dalgacık Dönüşümü ile EEG İşaretlerinden Çıkarılan Öznitelik Vektörleri Üzerinde İstatistiksel İşlemlerin Gerçekleştirilmesi, *Elektrik-Elektronik-Bilgisayar Mühendisliđi Sempozyumu ELECO-2004*, 230-234, Bursa.
- Güler, İ. ve Übeyli, E.D., 2006, Çok Katmanlı Perseptron Sinir Ağları ile Diyabet Diyabet Hastalığının Teşhisi, *Gazi Üniversitesi Müh. Mim. Fak. Dergisi*, Cilt 21, No 2, 319-326.
- Han, M. ve Sun, L., 2010, EEG Signal Classification for Epilepsy Diagnosis based on AR Model and RVM, *International Conference on Intelligent Control and Information Processing*, 13-15 Ağustos 2010 - Dalian, China.
- Harikumar, R. ve Narayanan, B. S., 2003, Fuzzy Techniques for Classification of Epilepsy Risk Level from EEG Signals, *TENCON 2003*, 209-213.
- Hema, C. R., Paulraj, M. P., Nagarajan, R., Yaacob, S. ve Adom, A. H., 2008, Application of Particle Swarm Optimization for EEG Signal Classification, *Biomedical Soft Computing and Human Sciences*, 13, No.1, 79-84.
- Kakıcı, A., 2009, <http://www.ahmetkakici.com/category/yapay-sinir-aglari/> [Ziyaret Tarihi: 12 Ağustos 2011].
- Kansu, O., 2006, Uzaktan Algılamada Görüntü Sınıflandırma Yöntemleri Analizi, *Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü*, Trabzon.
- Karabođa, D., 2011, Yapay Zeka Optimizasyon Algoritmaları, *Nobel Yayın Dağıtım*, Ankara.

- Kennedy, J. ve Eberhart, R., 1995, Particle Swarm Optimization, *Proceedings of IEEE International Conference on Neural Networks*, Avustralya, 1942-1948.
- Kulluk, S., 2009, Karınca Koloni Optimizasyonu ile Yapay Sinir Ağlarından Kural Çıkarımı, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü*, Kayseri.
- Liu, L., Liu, W., Cartes, D.A., 2008, Particle Swarm Optimization based Parameter Identification Applied to Permanent Magnet Synchronous Motors, *Engineering Applications of Artificial Intelligence*, 21, 1092-1100.
- Metz, C. E., 1978, Basic Principles of ROC Analysis, *Sem Nuc Med.*, 283-298.
- Ninomiya, H. ve Zhang, Q. J., 2008, Particle with Ability of Local Search Swarm Optimization:PALSO for Training of Feedforward Neural Networks, *2008 IEEE International Joint Conference on Neural Networks*, Hong Kong, 3009-3014.
- Özbeyaz, A., 2010, EMG ve EEG İşaretlerinin Sınıflandırılmasında EKK-DVM Hiper Parametrelerinin PSO' ya Dayalı Seçimi, *Kahramanmaraş Sütçü İmam Üniversitesi Fen Bilimleri Enstitüsü*, Kahramanmaraş.
- Öztemel, E., 2006, Yapay Sinir Ağları, *Papatya Yayıncılık*, İstanbul.
- Parsopoulos, K. E. ve Vrahatis, M. N., 2002, Recent Approaches to Global Optimization Problems Through Particle Swarm Optimization, *Natural Computing*, 1 (2-3), 235-306.
- Sağıroğlu, Ş., Beşdok, E. ve Erler, M., 2003, Mühendislikte Yapay Zeka Uygulamaları I Yapay Sinir Ağları, *Ufuk Kitap Kırtasiye-Yayıncılık Tic. Ltd. Şti.*, Kayseri.
- Sezer, E., 2008, Epilepsi Teşhisi için EEG Sinyal Analizi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*, Konya.
- Shi, Y. ve Eberhart, R. C., 1999, Empirical Study of Particle Swarm Optimization, *Congress on Evolutionary Computation*, Washington D.C., USA, 1945-1949.
- Sivasankari, N. ve Thanushkodi, K., 2009, Automated Epileptic Seizure Detection in EEG Signals Using FastICA and Neural Network, *Int. J. Advance. Soft Comput. Appl.*, 1, 2, 91-104.
- Subaşı, A., Alkan, A. ve Koklukaya, E., 2004, EEG İşaretlerinin Dalgacık Sinir Ağı ile Sınıflandırılması, *Teknoloji*, Cilt 7, Sayı 1, 71-80.
- Şahin, C., Oğulata, S. N., Aslan, K., Özdemir, H. ve Erol, R., 2008, A Neural Network-Based Classification Model for Partial Epilepsy by EEG Signals, *International Journal of Pattern Recognition and Artificial Intelligence*, 22, No. 5 (2008) 973-985.
- Şevkli, M., ve Yenisey, M. M., 2006, Atölye Tipi Çizelgeleme Problemleri için Parçacık Sürü Optimizasyonu Yöntemi, *İtÜdergisi/d Mühendislik*, Cilt 5, Sayı 2(1), 58-68.

- Tamer, S. ve Karakuzu, C., 2006, Parçacık Sürüsü Optimizasyon Algoritması ve Benzetim Örnekleri, *Kocaeli Üniversitesi Mühendislik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümü*, Kocaeli.
- Tewolde, G. S., ve Hanna, D. M., 2007, Particle Swarm Optimization for Classification of Breast Cancer Data using single and Multisurface Methods of Data Separation, *IEEE EIT 2007 Proceedings*, 443-446.
- Tezel, G., 2007, Biyomedikal İşaretlerin Yeni Bir Adaptif Aktivasyon Fonksiyonlu Yapay Sinir Ağları ile Sınıflandırılması, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*, Konya.
- Tezel, G. ve Özbay, Y. 2007, EEG Sinyallerinin Sınıflandırılmasında Yeni Bir Yaklaşım, *IEEE SİU 2007*, 11-13 Haziran 2007, Eskişehir.
- Toprak, İ. B., 2007, EEG Sinyallerinin Dalgacık Dönüşümü ve Yapay Sinir Ağları ile Analizi, *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü*, Isparta.
- Wang, Z., Sun, X. ve Zhang, D., 2007, A PSO-Based Classification Rule Mining Algorithm, *ICIC 2007*, LNAI 4682, 377-384.
- Yazgan, E. ve Korürek, M., 1996, *Tıp Elektronik. İTÜ*, No:15741, 386, İstanbul.
- Yetkin, M., 2008, GPS Ağlarının Optimal Tasarımı ve Robust İstatistik Yöntemlerin Kullanılabilirliği, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*, Konya.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Nesibe YALÇIN
Uyruğu : T.C.
Doğum Yeri ve Tarihi : Akdağmadeni – 04.11.1987
Telefon : 0537 8932198
e-mail : nesibeyalcin@hotmail.com

EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise	: Aydınlikevler İnönü (Y.D.A.) Lisesi, Altındağ, Ankara	2005
Üniversite	: Selçuk Üniversitesi M.M.F. Bilg. Müh. Bölümü, Selçuklu, Konya	2009
Yüksek Lisans:	Selçuk Üniversitesi F.B.E. Bilg. Müh. A.B.D., Selçuklu, Konya	

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2009	Belya A.Ş.	Yazılım Görevlisi
2011-...	Bilecik Üniversitesi	Araştırma Görevlisi

UZMANLIK ALANI: Web Yazılımcılığı, Veritabanı Yönetimi, Yapay Zeka

YABANCI DİLLER

İngilizce Okuma: İyi Yazma: İyi Konuşma: Orta Mesleki: İyi

DİĞER ÖZELLİKLER

YAYINLAR