

T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



LOJİSTİK REGRESYON ve ÇARPIMSAL
NÖRON MODELE DAYALI YENİ BİR
SINIFLANDIRICI

Erdoğan KOLAY

DOKTORA TEZİ

T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DOKTORA TEZİ

LOJİSTİK REGRESYON ve ÇARPIMSAL
NÖRON MODELE DAYALI YENİ BİR
SINIFLANDIRICI

Erdoğan KOLAY

İSTATİSTİK ANABİLİM DALI

SAMSUN
2017

Her Hakkı Saklıdır.

TEZ ONAYI

Erdiñ Kolay tarafından hazırlanan “Lojistik Regresyon ve Çarpımsal Nöron Modele Dayalı Yeni Bir Sınıflandırıcı” adlı tez çalışması .../.../2017 tarihinde aşağıdaki jüri tarafından Ondokuz Mayıs Üniversitesi Fen Bilimleri Enstitüsü İstatistik Anabilim Dalı’nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Danışman Doç. Dr. Taner TUNÇ
İstatistik Anabilim Dalı

Jüri Üyeleri

Başkan
..... Üniversitesi
..... Anabilim Dalı

Üye
..... Üniversitesi
..... Anabilim Dalı

Üye
..... Üniversitesi
..... Anabilim Dalı

Üye
..... Üniversitesi
..... Anabilim Dalı

Yukarıdaki sonucu onaylarım. .../.../2017

Prof. Dr. Bahtiyar ÖZTÜRK
Enstitü Müdürü

ETİK BEYAN

Ondokuz Mayıs Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez içindeki bütün bilgilerin doğru ve tam olduğunu, bilgilerin üretilmesi aşamasında bilimsel etiğe uygun davrandığımı, yararlandığım bütün kaynakları atıf yaparak belirttiğimi beyan ederim.

15.06.2017

Erdinç KOLAY

ÖZET

Doktora Tezi

LOJİSTİK REGRESYON ve ÇARPIMSAL NÖRON MODELE DAYALI YENİ BİR SINIFLANDIRICI

Erdoğan Kolay

Ondokuz Mayıs Üniversitesi
Fen Bilimleri Enstitüsü
İstatistik Anabilim Dalı

Danışman: Doç. Dr. Taner Tunç

Bu tez çalışmasında istatistiksel sınıflandırma problemlerinin çözümü için kullanılan ve istatistiksel bir teknik olan lojistik regresyon ile istatistiksel yöntemlere alternatif olarak geliştirilen yapay sinir ağı modellerinden biri olan çarpımsal nöron modelin birleştirilmesi ile yeni bir sınıflandırıcı modeli önerilmiştir. Önerilen yeni yapay sinir ağı sınıflandırıcısında, ağ eğitimi için evrimsel algoritmalarından bir tanesi olan Parçacık Sürü Optimizasyon algoritması yeniden uyarlanarak kullanılmıştır. Uyarlanmış Parçacık Sürü Optimizasyon algoritmasının performansının standart Parçacık Sürü Optimizasyon algoritmasından daha iyi performans gösterip göstermediği matematikte sıklıkla kullanılan karşılaştırma fonksiyonları yardımıyla test edilmiştir. Önerilen yeni yapay sinir ağı sınıflandırıcısının kestirim performansı, model seçim ölçütlerinden biri olan Çapraz Geçerlilik yönteminin gerçek hayat problemlerine uygulanması ile araştırılmıştır. Elde edilen sonuçlar literatürde sınıflandırma problemlerinin çözümü için sıklıkla kullanılan yapay sinir ağı modellerinden Çok Katmanlı Algılayıcı, Pi-Sigma Nöron Model ve Çarpımsal Nöron Model ile karşılaştırılmıştır. Elde edilen bulgular önerilen yeni yapay sinir ağı sınıflandırıcısının sınıflandırma problemlerinin çözümünde daha iyi kestirim performansı verdiğini göstermektedir.

Haziran 2017, 79 sayfa

Anahtar Kelimeler: İstatistiksel Sınıflandırma, Yapay Sinir Ağları, Parçacık Sürü Optimizasyonu, Çapraz Geçerlilik, Çarpımsal Nöron Model

ABSTRACT

Doctoral Dissertation

A NEW CLASSIFIER BASED ON LOGISTIC REGRESSION AND MULTIPLICATIVE NEURON MODEL

Erdoğan Kolay

Ondokuz Mayıs University
Graduate School of Sciences
Department of Statistics

Supervisor: Assoc.Prof. Dr. Taner Tunç

In this thesis a new hybrid classifier is proposed combining a statistical technique which is most commonly used for handling classification problems and multiplicative neuron model which is a type of Artificial Neural Network. For training the proposed neural network classifier, Particle Swarm Optimization is reorganized and applied. For testing performance of the adjusted Particle Swarm Optimization, benchmark functions which is commonly used for mathematical fields are used. For investigation of prediction performance of the proposed classifier, Cross Validation which is a criterion of the model selection is applied for real world datasets. Obtained results are compared with results of Multilayer Perceptron, Pi-Sigma Neural Network and Single Multiplicative Neuron Model which are used in classification problems in the literature. According to these results, the proposed neural network classifier has better prediction performance.

June 2017, 79 pages

Key Words: Statistical Classification, Artificial Neural Networks, Particle Swarm Optimization, Cross Validation, Multiplicative Neuron Model.

ÖNSÖZ VE TEŞEKKÜR

Doktora eğitimim süresince her türlü desteği sağlayan, çalışmalarına yön veren değerli danışmanım Sayın Doç. Dr. Taner Tunç'a gösterdiği yakınlık ve tüm katkılarından dolayı teşekkür ederim.

Doktora tez çalışmalarında yol gösteren, desteklerini esirgemeyen tez izleme komitesi üyeleri hocalarım Sayın Doç. Dr. Vedat Sağlam ve Sayın Doç. Dr. Mehmet Gürcan'a teşekkür ederim.

Tez çalışmam boyunca benden yardımlarını ve desteklerini esirgemeyen, değerli fikirleri ile çalışmalarına yön veren Sayın Prof. Dr. Erol Eğrioglu hocama teşekkür ederim.

Eğitim hayatımın tümünde benimle yakından ilgilenen, hayatım boyunca her zaman yanımda olan canım ablam Gülçin Kolay Bora'ya ve hayatımın tümünde bugünlere ulaşabilmem için her türlü cefayı çekmiş annem Ayşe Kolay'a ve rahmetli babam Ekrem Kolay'a tüm emekleri için şükranlarımı sunarım.

Doktora tez çalışmamın her aşamasında bana destek veren, sabır gösteren, teşvik eden sevgili eşim Burçak Kolay'a herşey için teşekkür ederim.

Doktora eğitimimin tüm aşamasında her türlü hususta destek sağlayan Sayın Yrd. Doç. Dr. Alper Sinan hocama ve son olarak aynı çatı altında görev yaptığımız değerli dostlarım Dr. Emre Dil ve Ömer Ferit İmlak'a ve Sinop Üniversitesi Fen Edebiyat Fakültesinde birlikte görev yaptığımız tüm arkadaşlara teşekkür ederim.

Haziran 2017, Samsun

Erdoğan Kolay

İÇİNDEKİLER DİZİNİ

| | |
|---|-----|
| ÖZET | i |
| ABSTRACT..... | ii |
| ÖNSÖZ VE TEŞEKKÜR | iii |
| İÇİNDEKİLER DİZİNİ | iv |
| SİMGELER VE KISALTMALAR..... | vi |
| ŞEKİLLER DİZİNİ..... | vii |
| ÇİZELGELER DİZİNİ | ix |
| 1. GİRİŞ | 1 |
| 1.1. Tezin Amacı | 3 |
| 2. LOJİSTİK REGRESYON ANALİZİ | 4 |
| 2.1. Lojistik Regresyon Modeli | 4 |
| 2.2. Parametre Tahmini | 5 |
| 3. YAPAY SİNİR AĞLARI | 9 |
| 3.1. Yapay Sinir Ağlarının Genel Özellikleri | 12 |
| 3.2. Yapay Sinir Ağının Ana Öğeleri | 12 |
| 3.2.1. Girişler | 13 |
| 3.2.2. Ağırlıklar..... | 13 |
| 3.2.3. Toplama işlemi | 13 |
| 3.2.4. Aktivasyon işlemi ve bazı aktivasyon fonksiyonları | 14 |
| 3.2.4.1. Eşik aktivasyon fonksiyonu | 14 |
| 3.2.4.2. Rampa aktivasyon fonksiyonu | 14 |
| 3.2.4.3. Lojistik aktivasyon fonksiyonu | 15 |
| 3.2.4.4. Tanjant Hiperbolik aktivasyon fonksiyonu | 15 |
| 3.3. Yapay Sinir Ağı Mimarileri | 16 |
| 3.3.1. Çok katmanlı algılayıcı | 16 |
| 3.3.2. Pi-Sigma nöron model | 21 |
| 3.3.3. Çarpımsal nöron model..... | 23 |
| 3.4. Yapay Sinir Ağı Eğitimi..... | 24 |
| 4. PARÇACIK SÜRÜ OPTİMİZASYONU | 29 |
| 4.1. Uyarlanmış Parçacık Sürü Optimizasyonu (UPSO)..... | 31 |
| 5. LOJİSTİK REGRESYON ve ÇARPIMSAL NÖRON MODELE DAYALI YENİ BİR SINIFLANDIRICI..... | 33 |
| 6. ÇAPRAZ GEÇERLİLİK | 37 |
| 7. BULGULAR ve TARTIŞMA..... | 40 |
| 7.1. PSO ve UPSO Algoritmalarının Karşılaştırılması | 40 |
| 7.2. Direkt Bölme Çapraz Geçerlilik Sonuçları | 45 |
| 7.3. 10-kat Çapraz Geçerlilik Sonuçları | 53 |
| 8. SONUÇ VE ÖNERİLER..... | 58 |

| | |
|-----------------|----|
| KAYNAKLAR | 60 |
| ÖZGEÇMİŞ | 67 |

SİMGELER VE KISALTMALAR

KISALTMALAR

| | |
|------|---|
| CCE | Çapraz Entropi Hata Fonksiyonu (Cross Entropy Error) |
| HKO | Hata Kareler Ortalaması |
| LR | Lojistik Regresyon |
| MLP | Çok Katmanlı Algılayıcı (Multilayer Perceptron) |
| PSO | Parçacık Sürü Optimizasyonu (Particle Swarm Optimization) |
| SMN | Çarpımsal Nöron Model (Single Multiplicative Neuron) |
| UPSO | Uyarlanmış Parçacık Sürü Optimizasyonu |
| YSA | Yapay Sinir Ağı |

ŞEKİLLER DİZİNİ

| | |
|--|----|
| Şekil 3.1. Basitleştirilmiş biyolojik sinir hücresi | 10 |
| Şekil 3.2. Yapay sinir hücresi..... | 13 |
| Şekil 3.3. Eşik aktivasyon fonksiyonu grafiği..... | 14 |
| Şekil 3.4. Rampa aktivasyon fonksiyonu | 15 |
| Şekil 3.5. Lojistik aktivasyon fonksiyonu | 15 |
| Şekil 3.6. Hiperbolik tanjant fonksiyonu..... | 16 |
| Şekil 3.7. Çok katmanlı YSA mimarisi..... | 17 |
| Şekil 3.8. MLP mimarisinin farklı bir gösterimi | 18 |
| Şekil 3.9. Pi-Sigma nöron model mimarisi | 22 |
| Şekil 3.10. Çarpımsal nöron model mimarisi | 23 |
| Şekil 3.11. Geri yayılım algoritması..... | 28 |
| Şekil 4.1. UPSO algoritması..... | 32 |
| Şekil 5.1. Lojistik regresyon ve çarpımsal nöron modele dayalı yeni sınıflandırıcı mimarisi | 35 |
| Şekil 6.1. Direkt bölme çapraz geçerlilik yöntemi | 37 |
| Şekil 6.2. K-kat çapraz geçerlilik | 38 |
| Şekil 7.1.. f1 fonksiyonu ve yakınsama grafiği | 42 |
| Şekil 7.2. f2 fonksiyonu ve yakınsama grafiği | 43 |
| Şekil 7.3. f3 fonksiyonu ve yakınsama grafiği | 43 |
| Şekil 7.4. f4 fonksiyonu ve yakınsama grafiği | 43 |
| Şekil 7.5. f5 fonksiyonu ve yakınsama grafiği | 44 |
| Şekil 7.6. f6 fonksiyonu ve yakınsama grafiği | 44 |
| Şekil 7.7. f7 fonksiyonu ve yakınsama grafiği | 44 |
| Şekil 7.8. f8 fonksiyonu ve yakınsama grafiği | 45 |

ÇİZELGELER DİZİNİ

| | |
|--|----|
| Çizelge 7.1. Karşılaştırma fonksiyonları..... | 41 |
| Çizelge 7.2. PSO ve UPSO performanslarının karşılaştırılması | 42 |
| Çizelge 7.3. UCI veri deposundan alınan orijinal sınıflandırma problemleri | 46 |
| Çizelge 7.4. Avustralya kredi onayı veri setinin eğitim performansının araştırılması | 47 |
| Çizelge 7.5. Meme kanseri teşhisi veri setinin eğitim performansının araştırılması | 48 |
| Çizelge 7.6. Şeker hastalığı veri setinin eğitim performansının araştırılması..... | 49 |
| Çizelge 7.7. Kalp hastalığı veri setinin eğitim performansının araştırılması..... | 50 |
| Çizelge 7.8. İyonosfer veri setinin eğitim performansının araştırılması | 51 |
| Çizelge 7.9. Meme kanseri veri setinin eğitim performansının araştırılması | 52 |
| Çizelge 7.10. Sağkalım süresi veri setinin eğitim performansının araştırılması | 53 |
| Çizelge 7.11. Avustralya kredi onayı problemi için 10-kat çapraz geçerlilik..... | 54 |
| Çizelge 7.12. Meme kanseri teşhisi problemi için 10-kat çapraz geçerlilik | 55 |
| Çizelge 7.13. Şeker hastalığı problemi için 10-kat çapraz geçerlilik..... | 55 |
| Çizelge 7.14. Kalp hastalığı problemi için 10-kat çapraz geçerlilik | 56 |
| Çizelge 7.15. Meme kanseri (orijinal) problemi için 10-kat çapraz geçerlilik | 56 |
| Çizelge 7.16. İyonosfer problemi için 10-kat çapraz geçerlilik..... | 57 |
| Çizelge 7.17. Sağkalım süresi problemi için 10-kat çapraz geçerlilik..... | 57 |

1. GİRİŞ

Bilimin en temel yöntemlerinden bir tanesi olan sınıflandırma, insanoğlu tarafından üstlenilen en eski bilimsel uğraşlardan biridir. Günümüzde bilim pek çok probleme çözüm üretmek amacıyla hızla ilerlemektedir. Sınıflandırma problemleri de bilimin çözüm üretmeye çalıştığı ve sıklıkla karşılaşılan bir problem türüdür.

Sınıflandırma, araştırmacının bir birey (denek, birim) üzerinde ölçümler yaptığında, bu ölçümlere bağlı olarak bireyi bilinen gruplardan birine atanmasını gerçekleştiren istatistiksel bir teknik olarak tanımlanır (Atakan, 1997).

Birçok alanda başarılı çalışmalar gerçekleştirebilmek için sınıflandırma problemlerinin etkin şekilde çözülmesine ihtiyaç duyulmaktadır. Bu nedenle, farklı alanlardan birçok araştırmacı, sınıflandırma problemi üzerinde çalışmakta ve daha iyi sonuçlar elde etmek için yeni yöntem arayışları sürmektedir (Örkcü, 2009).

Sınıflandırma problemlerinin çözümüne yönelik çalışmalar oldukça eskiye dayanır. İlk kez Fisher (1936) tarafından varyans analizinde gruplar arası ayrımsamayı maksimize etmek amacı ile iki grup için önerilen diskriminant analizi, bilinen en eski sınıflandırma yöntemidir. Diskriminant analizi yönteminin temel varsayımları, değişkenlerin çok değişkenli normal dağılması ve grupların ortak varyans-kovaryans matrisli olmalarıdır (Rencher, 2002; Izenmann, 2008).

Diskriminant analizinde varsayımlar bozulduğunda lojistik regresyon analizi önerilmektedir. Lojistik regresyon açıklayıcı değişkenlerin kesikli olduğu durumda da uygulanabildiği için diskriminant analizinden daha güçlü kestirimler vermekte dolayısıyla daha iyi modele götürebilmektedir (Knoke, 1982; Başarır, 1990).

Alternatif yöntem olarak geliştirilen ve genelleştirilmiş lineer modeller ailesinin özel bir durumu olan lojistik modeli araştıran lojistik regresyon analizi medikal uygulamalarda, özellikle epidemiyolojide, ekonomide ve sınıflandırma problemlerinde sıklıkla kullanılmaktadır. Çözümlemede elde edilen modelin matematiksel olarak çok esnek olması ve kolay yorumlanabilmesi yönetime olan ilginin artmasına neden olmuştur (Tatlıdil, 1996).

Sınıflandırma, regresyon, kümeleme vb. problem tipleri için birçok istatistiksel yöntem geliştirilmiştir. Bu yöntemlerin bazıları varsayımların sağlandığı durumlarda başarılı sonuçlar vermekle birlikte, tasarlandıkları gözlem kümesinin haricindeki gözlemler için iyi performans gösterecek kadar esnek değildirler. Yapay Sinir Ağları (YSA) klasik yöntemlere alternatif bir yaklaşım olarak kullanılmaktadır (Jain vd., 1996).

Biyolojik sinir sisteminin matematiksel olarak modellenmesi sonucu ortaya çıkan YSA, paralel çalışma ve öğrenebilme yetenekleri bakımından biyolojik sinir sisteminin özelliğini göstermektedir. Diğer özelliklerinin yanında paralel çalışmasından dolayı bilgileri hızlı bir şekilde işleyebilmesi ve donanımının kolayca gerçekleştirilebilir olması YSA'yı başka yöntemlere göre daha cazip kılmaktadır. Bu yüzden YSA; sınıflandırma, mühendislik uygulamaları, kontrol, zaman serisi kestirimi, ses, yüz ve görüntü tanıma vb. birçok alanda yoğun olarak kullanılmaktadır. YSA'lar birçok alanda yeni kullanılmaya başlanmasına rağmen her geçen gün kullanım alanları gelişerek artmaktadır (Şen, 2004).

YSA; birbirlerine paralel olarak çalışan, birçok basit işlem elemanından oluşan ve fonksiyonu, ağıın yapısı, bağlantı ağırlıkları ve elemanlarda gerçekleştirilen işlemler tarafından belirlenen bir sistemdir (Widrow vd. 1988).

YSA'nın sınıflandırma probleminde kullanılmasındaki hedef, verilen bir veri kümesindeki hatalı sınıflandırma oranını yani hatayı en aza indirgeyecek ağıın bulunmasıdır. En iyi ağıın bulunması aynı zamanda Hata Kareler Ortalaması (HKO) ya da benzeri bir hata fonksiyonunun (Çapraz Entropi, Dayanıklı hata fonksiyonları) en küçüklenmesi problemi olarak tanımlanabilir. Hata fonksiyonunun en küçüklenmesi için ağıın parametre değerlerinin ayarlanması işlemine ağıın eğitimi denir. Ağıın eğitimi yani YSA parametrelerinin (ağırlık, eşik, gizli tabaka sayısı) değerlerinin bulunması bir optimizasyon problemi olarak da düşünülebilir.

Parçacık sürü optimizasyonu (PSO), Eberhart ve Kennedy (1995) tarafından önerilmiş evrimsel bir optimizasyon algoritmasıdır. PSO, örneklem temelli, stokastik ve sezgisel bir algoritmadır. Doğadaki kuş, balık vb. hayvanların sosyal davranışlarının benzetimi olarak düşünülmüştür. PSO'da yer alan her bir eleman parçacık diye adlandırılır ve bu parçacıkların her biri arama uzayında en iyi çözüm

için adaydır. Tüm parçacıkların oluşturduğu sürü, adımsal süreçte her bir parçacığın en iyi geçmiş deneyimini saklar ve bir sonraki adımda kullanır.

1.1. Tezin Amacı

Bu çalışmada sınıflandırma problemlerinin çözümünde yüksek sınıflandırma performansı elde edebilmek için lojistik regresyon ve YSA çeşitlerinden biri olan çarpımsal nöron model birleştirilerek yeni bir YSA modeli önerilmiştir. Önerilen yeni YSA modelinde ağ eğitimi için PSO tekrardan düzenlenerek, uyarlanmış PSO kullanılmıştır. Önerilen yeni eğitim algoritmasının performansı matematikte sıklıkla kullanılan karşılaştırma fonksiyonları üzerinde test edilmiştir. Ayrıca tezin konusunu oluşturan yeni sinir ağı sınıflandırıcısı gerçek hayat problemlerine uygulanmış ve sonuçlar literatürde sıkça kullanılan ileri beslemeli çok katmanlı algılayıcı, Pi-Sigma Nöron model ve Çarpımsal Nöron model ile karşılaştırılmıştır.

2. LOJİSTİK REGRESYON ANALİZİ

Regresyon yöntemleri bağımlı değişken(ler) ile bağımsız değişken(ler) arasındaki ilişkinin araştırıldığı istatistiksel yöntemlerdir. Bağımlı değişkenin kesikli olduğu durumlarla sıklıkla karşılaşılır. Son yıllarda böyle durumlar için lojistik regresyon birçok alanda kullanılır hale gelmiştir (Hosmer ve Lemeshow, 2000).

2.1. Lojistik Regresyon Modeli

Lojistik regresyon analizinde amaç birimin bilinen gruplardan hangisinde yer aldığına dair olasılık modeli elde etmektir. Elde edilen bu model ile birimin gruplara ait olma olasılıkları hesaplanır ve yüksek olasılığa sahip gruba atama işlemi yapılır. Lojistik regresyon modelini anlamak için öncelikle genel doğrusal regresyon modelinin incelenmesi gerekir. Genel doğrusal regresyon modelini (2.1)'deki gibi koşullu beklenen değer biçiminde yazmak mümkündür.

$$E(y_i / x_{i1}, \dots, x_{ip}) = \sum_{k=0}^K \beta_k x_{ik}; \quad i = 1, 2, \dots, n \quad (2.1)$$

Bu modelde bağımsız değişkenler üzerinde kısıt yok iken, y bağımlı değişkeninin sürekli olması koşulu vardır. Bu model verilen her i gözlemi için,

$$y_i = \sum_{k=0}^K \beta_k x_{ik} + u_i \quad (2.2)$$

olarak ifade edilebilir (Burada $x_{i0} = 1$ 'dir.). Modelde bağımsız değişkenler üzerinde herhangi bir kısıt olmadığından y_i bağımlı değişkeni değeri $(-\infty, +\infty)$ arasında tüm değerleri alabilmektedir (Başarı, 1990).

Bağımlı değişkenin (ikili yanıt değişkeni);

$$y_i = \begin{cases} 1, & \text{İzleme periyodu boyunca olay meydana gelmişse} \\ 0, & \text{Diğer durumlarda} \end{cases}$$

şeklinde 0,1 gibi kategorik değerler aldığı durumlarda bu kural bozulmakta ve $P(y_i = 1)$, i. gözlemin 1 değerini alma olasılığı olmak üzere beklenen değer;

$$E(y_i) = 1.P(y_i = 1) + 0.P(y_i = 0) = P(y_i = 1) \quad (2.3)$$

olmaktadır. Bu sonuç regresyon denklemi olarak yazılırsa,

$$E(y_i) = P(y_i = 1) = \sum_{k=0}^K \beta_k x_{ik} \quad (2.4)$$

sonucu elde edilir. Eşitlik (2.4) ile verilen doğrusal olasılık modelinde denklemin sol tarafı $[0,1]$ aralığında sınırlı olasılık değerleri alır. Bağımsız değişkenler ise sonsuz değerler alabildiği için eşitlik her zaman sağlanamayabilir. Logit dönüşüm kullanılarak (2.4) ile verilen model üzerinde $\frac{P}{1-P}$ dönüşümü ile bağımlı değişkenin sınırları $[0, \infty]$ arasına ve daha sonra doğal logaritma alınarak bağımlı değişkenin sınırları $[-\infty, +\infty]$ arasına getirilmektedir. Dönüşüm sonucu yeni fonksiyon,

$$E(y_i) = L_i = \log(P_i / (1 - P_i)) = \sum_{k=0}^K \beta_k x_{ik} \quad (2.5)$$

olarak yazılmaktadır. Eşitlik (2.5) ile verilen denklemin her iki tarafı e tabanında yazılırsa;

$$\frac{P_i}{1 - P_i} = e^{\sum_{k=0}^K \beta_k x_{ik}} \quad (2.6)$$

elde edilir. Lojistik model olarak bilinen bu modelde P_i olasılık değeri

$$P_i = \frac{e^{\sum_{k=0}^K \beta_k x_{ik}}}{1 + e^{\sum_{k=0}^K \beta_k x_{ik}}} = \frac{1}{1 + e^{-\sum_{k=0}^K \beta_k x_{ik}}} \quad (2.7)$$

şeklinde tanımlanır. Eşitlik (2.7) lojistik fonksiyon olarak bilinir ve bu P_i olasılıklarının grafiği “S” şeklinde olup, bu grafiğe lojistik fonksiyon eğrisi denir (Başarır, 1990).

2.2. Parametre Tahmini

Lojistik regresyonda amaç (2.5) ile verilen $K=(p+1)$ tane β parametresinin tahmin edilmesidir. Gözlemlenen verinin olasılığını en yüksek yapan parametre kümesinin bulunmasını sağlayan en çok olabilirlik yöntemi ile parametre değerleri elde edilir. En çok olabilirlik denklemi bağımlı değişkenin dağılımından elde edilir. Her i-inci

popülasyonda y_i ikili sonuçlu olduğundan, Y 'ye ait ortak olasılık yoğunluk fonksiyonu,

$$f(y \setminus \beta) = \prod_{i=1}^n \frac{n_i!}{y_i!(n_i - y_i)!} p^{y_i} (1-p)^{n_i - y_i} \quad (2.8)$$

şeklindedir. Olabilirlik fonksiyonu olasılık yoğunluk fonksiyonu ile aynı formdadır.

$$f(\beta \setminus y) = \prod_{i=1}^n \frac{n_i!}{y_i!(n_i - y_i)!} p^{y_i} (1-p)^{n_i - y_i} \quad (2.9)$$

En çok olabilirlik tahminleri (2.9)'un maksimize edildiği β değerleridir. Fonksiyonun ilk türevi sıfıra eşit olduğunda kritik noktaları (minimum ya da maksimum) mevcuttur. Eğer ikinci türev değeri negatif ise o zaman kritik nokta maksimumdur. Bu nedenle en çok olabilirlik tahminlerinin elde edilmesi için birinci ve ikinci türevin hesaplanmasına ihtiyaç vardır. Eşitlik (2.9)'da verilen denklemin β 'ya göre türevinin hesaplanması çarpım terimlerinin karmaşıklığı nedeni ile zordur. Ancak olabilirlik denklemi daha kolay forma dönüştürülebilir. İlk olarak, faktöriyel içeren kısım β değeri içermemektedir. Bu nedenle bu kısım sabit olarak düşünülebilir ve ihmal edilebilir. Çünkü olabilirlik fonksiyonunu faktöriyel kısım olmadan maksimum yapmak, bu terimlerin varlığında yapılan işlem ile aynı sonucu verir. İkinci olarak $a^{x-y} = a^x / a^y$ kullanılarak ve terimler yeniden düzenlenerek maksimum yapılması gereken olabilirlik denklemi (2.10)'daki gibi yazılabilir.

$$f(\beta \setminus y) = \prod_{i=1}^N \left(\frac{p_i}{1-p_i} \right)^{y_i} (1-p_i)^{n_i} \quad (2.10)$$

Eşitlik (2.10)'un ilk terimi (2.6)'da ve ikinci terimi (2.7)'de verildiği gibi yazılarak basit matematiksel işlemler sonucunda olabilirlik denklemi (2.11)'deki gibi yazılabilir.

$$f(\beta \setminus y) = \prod_{i=1}^N \left(e^{y_i \sum_{k=0}^{p+1} x_{ik} \beta_k} \right) \left(1 + e^{\sum_{k=0}^{p+1} x_{ik} \beta_k} \right)^{-n_i} \quad (2.11)$$

Ancak bu eşitlik hala türev işlemi için elverişsizdir. Logaritma monoton bir fonksiyon olduğu için olabilirlik fonksiyonunun maksimum değeri log-olabilirlik fonksiyonunki ile aynı olacaktır. Log-Olabilirlik fonksiyonu (2.12) ile gösterilebilir.

$$l(\beta) = \sum_{i=1}^N y_i \left(\sum_{k=0}^{p+1} x_{ik} \beta_k \right) - \sum_{i=1}^N n_i \log \left(1 + e^{\sum_{k=0}^{p+1} x_{ik} \beta_k} \right) \quad (2.12)$$

Log-olabilirlik fonksiyonun kritik noktalarının bulunması için β_k 'ya göre birinci türevi alınarak sıfıra eşitlenir. Burada $\frac{\partial}{\partial \beta_k} \sum_{k=0}^K x_{ik} \beta_k = x_{ik}$ olmak üzere en çok olabilirlik denklemi aşağıdaki şekilde elde edilir.

$$\begin{aligned}
\frac{\partial l(\beta)}{\partial \beta_k} &= \sum_{i=1}^N y_i x_{ik} - \sum_{i=1}^N n_i \frac{1}{1 + e^{\sum_{k=0}^{p+1} x_{ik} \beta_k}} \frac{\partial}{\partial \beta_k} (1 + e^{\sum_{k=0}^{p+1} x_{ik} \beta_k}) \\
&= \sum_{i=1}^N y_i x_{ik} - \sum_{i=1}^N n_i \frac{1}{1 + e^{\sum_{k=0}^{p+1} x_{ik} \beta_k}} e^{\sum_{k=0}^{p+1} x_{ik} \beta_k} \frac{\partial}{\partial \beta_k} \sum_{k=0}^{p+1} x_{ik} \beta_k \\
&= \sum_{i=1}^N y_i x_{ik} - \sum_{i=1}^N n_i \frac{1}{1 + e^{\sum_{k=0}^{p+1} x_{ik} \beta_k}} \sum_{k=0}^{p+1} x_{ik} \\
&= \sum_{i=1}^N y_i x_{ik} - \sum_{i=1}^N n_i p_i x_{ik} \tag{2.13}
\end{aligned}$$

β 'nın en çok olabilirlik tahminleri (K+1) tane denklemin her birinin sıfıra eşitlenerek her bir β_k için çözümü ile elde edilir. Bu tip her çözüm (şayet çözüm varsa) maksimum veya minimum olmak üzere bir kritik nokta belirtir. Eğer ikinci kısmi türevler matrisi negatif tanımlı ise; yani köşegen matrisinin her elemanı sıfırdan küçük ise kritik nokta maksimum olacaktır. Bu matrisin bir diğer önemli özelliği de parametre tahminlerinin varyans-kovaryans matrisini oluşturmasıdır. Bu matris (2.13)'deki K+1 denklemin her birinin β_k ile gösterilen her β elemanına göre ikinci türevinin alınmasıyla oluşturulur. İkinci kısmi türevler matrisinin genel formu (2.14) ile verilmektedir.

$$\begin{aligned}
\frac{\partial^2 l(\beta)}{\partial \beta_k \partial \beta_{k'}} &= \frac{\partial}{\partial \beta_{k'}} \sum_{i=1}^N y_i x_{ik} - n_i x_{ik} \pi_i \\
&= \frac{\partial}{\partial \beta_{k'}} \sum_{i=1}^N -n_i x_{ik} \pi_i \\
&= - \sum_{i=1}^N n_i x_{ik} \frac{\partial}{\partial \beta_{k'}} \left(\frac{e^{\sum_{k=0}^K x_{ik} \beta_k}}{1 + e^{\sum_{k=0}^K x_{ik} \beta_k}} \right) \tag{2.14}
\end{aligned}$$

Denklemin çözümü için bilinen iki türev kuralı uygulanır.

$$\frac{d}{dx} e^{u(x)} = e^{u(x)} \frac{d}{dx} u(x) \quad (2.15)$$

Burada $u(x) = \sum_{k=0}^K x_{ik} \beta_k$ olsun. Kullanılacak ikinci türev kuralı ise türevde bölüm kuralıdır.

$$\left(\frac{f}{g} \right)'(a) = \frac{g(a)f'(a) - f(a)g'(a)}{[g(a)]^2} \quad (2.16)$$

(2.15) ve (2.16)'da verilen kurallar (2.14)'e uygulanarak

$$\begin{aligned} \frac{d}{dx} \frac{e^{u(x)}}{1+e^{u(x)}} &= \frac{(1+e^{u(x)}) \cdot e^{u(x)} \frac{d}{dx} u(x) - e^{u(x)} \cdot e^{u(x)} \frac{d}{dx} u(x)}{[1+e^{u(x)}]^2} \\ &= \frac{e^{u(x)} \frac{d}{dx} u(x)}{(1+e^{u(x)})^2} \\ &= \frac{e^{u(x)}}{1+e^{u(x)}} \frac{1}{1+e^{u(x)}} \frac{d}{dx} u(x) \end{aligned}$$

elde edilir. Böylece (2.14)'ün çözümü (2.17)'deki gibi yazılabilir.

$$\frac{\partial^2 l(\beta)}{\partial \beta_k \partial \beta_{k'}} = - \sum_{i=1}^N n_i x_{ik} p_i (1-p_i) x_{ik'} \quad (2.17)$$

Yukarıda verilen (2.13) denkleminin sıfıra eşitlenmesi ile (K+1) bilinmeyenli (K+1) lineer olmayan denklem sistemi elde edilir. Sistemin çözümü β_k elemanlarını içeren bir vektördür. İkinci kısmi türevler matrisinin negatif tanımlı olduğu ve çözümün yerel maksimumdan ziyade küresel maksimum olduğu doğruladıktan sonra, bu vektörün gözlemlenen verinin meydana gelme olasılığını en yüksek yapacak şekilde parametre tahminlerini içerdiği sonucu çıkarılır. Buna rağmen lineer olmayan denklem sistemini çözmek kolay değildir. Çözüm, lineer denklem sistemlerindeki gibi cebirsel olarak elde edilemez ancak adımsal bir süreç kullanılarak nümerik olarak tahmin edilebilir (Czepiel, 2002).

En iyi parametre değerlerinin belirlenmesi için Newton Raphson yöntemi gibi optimizasyon yöntemleri kullanılabilir.

3. YAPAY SİNİR AĞLARI

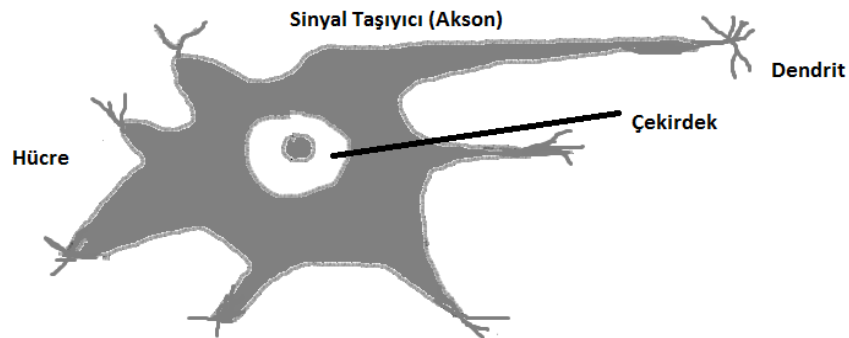
İnsanoğlu, herhangi bir aktivite sırasında oldukça karmaşık biyolojik sinir ağlarını kullanır. İnsanlar; okuma, nefes alma, düşünme vb. aktiviteleri gerçekleştirebilmek için birbirine bağlı yaklaşık 10^{11} biyolojik sinir ağına sahiptir. Her bir biyolojik nöron (sinir hücresi) oldukça zengin bir kimyaya ve dokuya sahiptir. Sinirsel yapıların bazıları doğumdan itibaren bizimledir, bazıları ise deneyimlerle gelişir.

Bilim adamları basitleştirilmiş bir biyolojik sinir hücresinin nasıl çalıştığını anlamaya çalışmışlardır. Genellikle tüm biyolojik sinir fonksiyonları nöronlarda ve nöronlar arasındaki bağlantılarda saklanır. Öğrenme, yeni bağlantıların kurulması ya da mevcut bağlantıların gelişmesi ile sağlanır. Bilim adamları, sinir ağlarının çalışma sistemi tam olarak bilinmese de yapay bir sinir ağı yapısının oluşturulmasını ve bunların fonksiyon yardımıyla eğitilmesinin mümkün olacağını öne sürmüşlerdir (Hagan vd., 1996).

YSA'lar günümüzde biyolojik sinir sistemi ile ilgili olarak bilinen bazı bulguların ayrıntılarının ihmal edilerek teknoloji ve bilimsel araştırma yöntemlerine uygulanmasıdır. YSA birbiri ile paralel iletişim içinde bulunan tabakalara sahip olan ve her bir tabakasında yeterli sayıda sinir hücresi bulunan bir sistemden oluşmaktadır. Bu tabakalar ve onların hücreleri arasında oldukça karmaşık sayılabilecek iletişimler, incelenen olayın yapısına göre belirlenen ağırlık katsayıları ile temin edilmektedir.

Biyolojik sinir sisteminin modellenmesi sonucu ortaya çıkan YSA'lar, paralel çalışma ve öğrenebilme yetenekleri bakımından biyolojik sinir sisteminin özelliğini göstermektedir. Diğer özelliklerinin yanında paralel çalışmasından dolayı bilgileri hızlı bir şekilde işleyebilmesi ve donanımının kolayca gerçekleştirilebilir olması YSA'yı başka yöntemlere göre daha cazip kılmaktadır. Bu yüzden YSA; sınıflandırma, desen tanıma, kontrol, görüntü işleme, ses tanıma vb. birçok alanda yoğun olarak kullanılmaktadır. YSA'lar birçok alanda yeni kullanılmaya başlamasına rağmen her geçen gün kullanım alanları gelişerek artmaktadır (Şen, 2004).

YSA üzerindeki ilk çalışma, McCullogh ve Pitts (1943)'in YSA'nın tanımını yaparak hücre modelini geliştirmesi olarak kabul edilir. Yapılan bu çalışmada sinir hücreleri sabit eşik değerli mantıksal elemanlar olarak modellenmiştir. Hebb (1949), YSA'daki öğrenme için başlangıç noktası sayılabilecek bir kural geliştirmiştir. Ortaya atılan bu öğrenme kuralı, o dönemde bir sinir ağının öğrenme işini nasıl gerçekleştirebileceği konusunda fikir vermekle birlikte, bugün halen geçerli olan öğrenme kurallarından birçoğunun da temelini oluşturmuştur. YSA konusunda ikinci büyük çıkış, Rosenblatt (1958) tarafından tekli doğrusal algılayıcı (perceptron) modeli ve öğrenme kuralı ile yapılmıştır. Önerilen bu model aynı zamanda bugünkü makine öğrenme algoritmasının temeli olmuştur. 1959 yılında, Bernard Widrow ve MarciaHoff "ADALINE" ve "MADALINE" diye adlandırdıkları ağ modellerini geliştirdiler. MADALINE, telefon hatlarında oluşan yankıları yok eden bir uyarlanabilir süzgeç olarak kullanılmış, gerçek dünya sorunlarına uygulanmış olan ilk sinir ağıdır ve hala kullanımda bulunmaktadır. Minsky ve Papert 1969, Perceptron'un yetersizliğini görmüşler ve XOR problemlerini çözemediğini ispatlamışlardır. Bunun için iki katmanlı ileri beslemeli ağların kullanılabilirliğini ileri sürmüşler ve tek katmanlı ağlardaki birçok sınırlamayı ortadan kaldırdığını göstermişlerdir. Fakat gizli katmanların ağırlıklarının nasıl değiştirileceği konusunda bir yöntem önermemişlerdir. Bu soruna Rumelhart vd 1986 çalışmalarında, geri yayılım algoritması ile bir çözüm getirmişlerdir (Şen, 2004; Elmas, 2011). Geri yayılım algoritmasının YSA'ya başarılı ile uygulanmasından sonra, YSA birçok bilimsel problemin çözümü için kullanılmıştır. Zhang (2000), çalışmasında 2000'li yıllara kadar olan çalışmaları özetlemiştir.



Şekil 3.1. Basitleştirilmiş biyolojik sinir hücresi (Şen, 2004)

YSA, Şekil 3.1 ile verilen basit biyolojik sinir hücrelerinden oluşan sisteminin çalışma şeklinin benzetimi için tasarlanan programdır. Benzetimi gerçekleştirilen nöronlar çeşitli şekillerde birbirlerine bağlanarak bir ağ oluştururlar. Bu ağlar; öğrenme, hafızaya alma ve veriler arasındaki ilişkiyi ortaya çıkarma kapasitesine sahiptirler. Diğer bir ifadeyle YSA'lar, normalde bir insanın düşünme ve gözlemlemeye yönelik doğal yeteneklerini gerektiren problemlere yönelik çözüm üretmektedir. Bir insanın, düşünme ve gözleme yeteneklerini gerektiren problemlere yönelik çözümler üretebilmesinin temel sebebi ise insan beyninin ve dolayısıyla insanın sahip olduğu yaşayarak ve deneyerek öğrenme yeteneğidir. Bir nöron; diğer sinir hücrelerinden gelen uyarıları sinapsları üzerinden dendritlere alır. Dendritler sinyalleri hücre gövdesine iletirler. Hücre de gelen sinyalleri birbirlerini kuvvetlendirme ve zayıflatma etkilerine göre işlerler. Eğer sinyaller bir eşik değerini aşabilirlerse aksonlara yollanır aksi halde sinyal gönderilmez. Bu aksonlar bilginin tutulduğu yerdir. YSA kavramı, işte bu çalışma sisteminin bilgisayarlar üzerinde matematiksel olarak taklit edilmesi fikri ile ortaya çıkmıştır (Kubat, 2012).

İnsan beyninin fonksiyonel özelliklerine benzer şekilde,

- Öğrenme
- İlişkilendirme
- Sınıflandırma
- Genelleme
- Özellik Belirleme
- Optimizasyon

gibi konularda başarılı şekilde uygulanmaktadır. Örneklerden elde ettikleri bilgiler ile kendi deneyimlerini oluşturur ve daha sonra benzer konularda benzer kararları verirler.

Teknik olarak bir YSA'nın en temel görevi, kendisine gösterilen bir girdi setine karşılık gelebilecek bir çıktı seti belirlemektir. Bunu yapabilmesi için ağ, ilgili olayın örnekleri ile eğitilerek (öğrenme) genelleme yapabilecek yeteneğe kavuşturulur. Bu genelleme ile benzer olaylara karşılık gelen çıktı setleri belirlenir (Öztemel, 2012).

3.1. Yapay Sinir Ağlarının Genel Özellikleri

YSA'da bilgilerin işlenmesi paralel olarak gerçekleştirildiğinden, taşınan bilgiler birbirinden bağımsızdır. Ayrıca aynı tabakadaki bağlantılar arasında zaman bağımlılığı olmadığından tamamı ile eş zamanlı çalışabilmekte dolayısıyla da bilgi akış hızı artmaktadır. Paralel çalışma ilkesinden dolayı herhangi bir birimde meydana gelecek hata tüm sistemi etkilemez. Yalnız hücrenin ağırlığı oranında bir etkilenme meydana gelebilir. Böylece genel sistem yerel hatalardan en az şekilde etkilenmektedir (Şen, 2004).

YSA'ların temelinde makine öğrenimi yatmaktadır. Bir YSA'ya verilen örnekler ile ağı eğitilmesi sağlanır. Eğitilen ağ karşısına gelecek yeni örnekler için geçmişte öğrendiği bilgiyi kullanarak sonuç üretir. Örneğin; Wisconsin Üniversitesinde meme kanseri teşhisi için yapılan bir çalışmada, görüntüleme sistemleri ile kişilerden alınan 30 değişkenli bir örnek veri seti ağ eğitiminde kullanılmış ve YSA'ların meme kanseri teşhisinde %96-%97'lik doğru sınıflandırma yapabildiği görülmüştür (Izenmann, 2008).

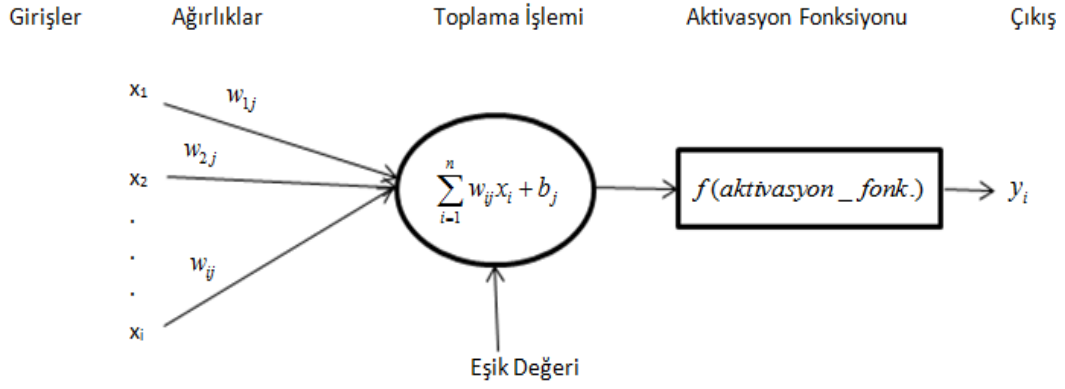
3.2. Yapay Sinir Ağının Ana Öğeleri

YSA'lar, birbirine bağlı çok sayıda işlem elemanından oluşmuş, genellikle paralel işleyen yapılar olarak adlandırılabilirler.

YSA'ların temel birimi işlem elemanı ya da düğüm olarak adlandırılan yapay bir sinir hücresidir. Bir yapay sinir hücresi, biyolojik sinirlere göre daha basit olmasına karşın, biyolojik sinirlerin 4 temel işlevini taklit ederler. Bu işlevler;

- Girişler
- Ağırlıklar
- Toplama İşlemi
- Aktivasyon İşlemi

şeklindedir. Şekil 3.2'de yapay sinir hücresinin genel hali görülmektedir.



Şekil 3.2. Yapay sinir hücresi

YSA'ların temel çalışma mantığı Şekil 3.2 ile verilen girişlerin her birinin ağırlık değerleri ile çarpılır ve eşik değeri ile toplanmasıdır.

3.2.1. Girişler

Girişler ya da Girdiler (x_1, x_2, \dots, x_n) dışarıdan alınan bilgi ya da veri olarak tanımlanabilir.

3.2.2. Ağırlıklar

Ağırlıklar (w_1, w_2, \dots, w_n), girdi değerlerinin toplam üzerindeki etkisini belirleyen katsayılardır. Her bir giriş değeri kendine ait bir ağırlığa sahiptir.

Bir ağırlığın değerinin büyük olması, o girişin toplama etkisinin yüksek olduğunu dolayısıyla önemli olduğunu gösterir.

3.2.3. Toplama işlemi

Toplama işlemi, hücrede her bir ağırlığın ait olduğu girdilerle çarpımının toplamlarının eşik değeri ile toplanması şeklinde yapılır. Bazı durumlarda toplama işlemi yerine minimum, maksimum, çarpım vb. operatörler kullanılabilir. Herhangi bir sinir hücresi için toplam (3.1)'deki gibi hesaplanır.

$$net = \sum_{i=1}^n w_{ij}x_i + b_j \quad (3.1)$$

3.2.4. Aktivasyon işlemi ve bazı aktivasyon fonksiyonları

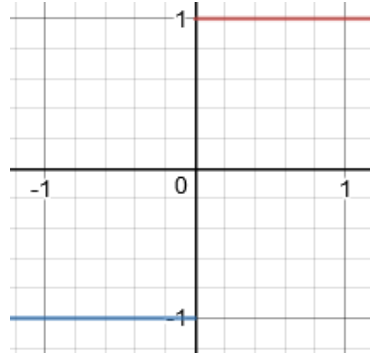
Toplama işleminin sonucu, aktivasyon fonksiyonu yardımıyla çıkışa iletilir. En çok kullanılan aktivasyon fonksiyonları aşağıda verilmiştir. Sınıflandırma problemleri için literatürde en çok kullanılan aktivasyon fonksiyonu lojistik aktivasyon fonksiyonudur.

3.2.4.1. Eşik aktivasyon fonksiyonu

Bu aktivasyon fonksiyonu ile tüm girdi değerlerine karşın yalnız iki çeşit (-1 ve 1) çıktı değeri üretilir. Matematiksel olarak,

$$f(net) = \begin{cases} 1 & net \geq 0 \\ -1 & net < 0 \end{cases}$$

şeklinde ifade edilen, eşik aktivasyon fonksiyonu grafiksel olarak Şekil 3.3'de gösterilmiştir.



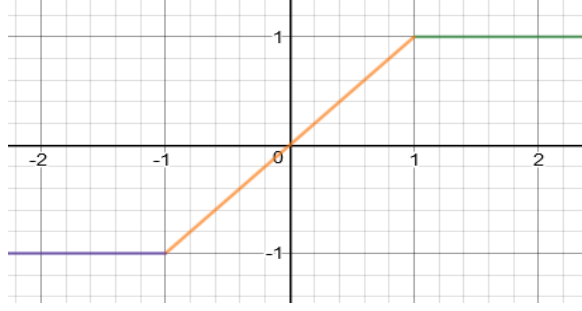
Şekil 3.3. Eşik aktivasyon fonksiyonu grafiği

3.2.4.2. Rampa aktivasyon fonksiyonu

Bir sinir hücresinin aktivasyonunda bu fonksiyon kullanırsa, aşağıda verilen net değerlerine göre çıkış değerleri elde edilir. Matematiksel gösterimi,

$$f(net) = \begin{cases} 1 & net > 1 \\ net & -1 \leq net \leq 1 \\ -1 & net < -1 \end{cases}$$

şeklinde dir. Rampa aktivasyon fonksiyonunun grafiksel gösterimi ise Şekil 3.4'deki gibi olacaktır.



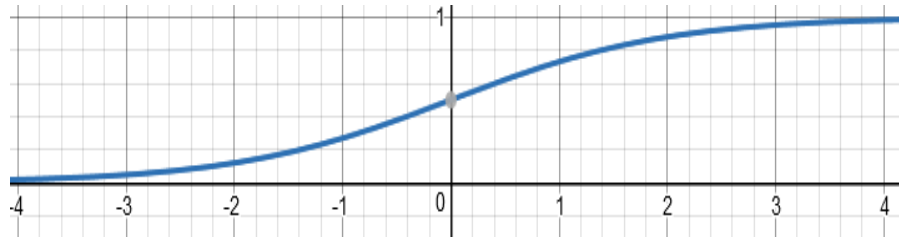
Şekil 3.4. Rampa aktivasyon fonksiyonu

3.2.4.3. Lojistik aktivasyon fonksiyonu

Lojistik aktivasyon fonksiyonu lineer olmayan bir fonksiyon olup, sınıflandırma problemleri için literatürde en sık kullanılan aktivasyon fonksiyonudur. Fonksiyon grafiği “S” şeklinde olup, matematiksel ifadesi

$$f(net) = \frac{1}{1 + e^{-net}}, -\infty < net < \infty$$

iken, fonksiyon grafiği Şekil 3.5’de verilmiştir.



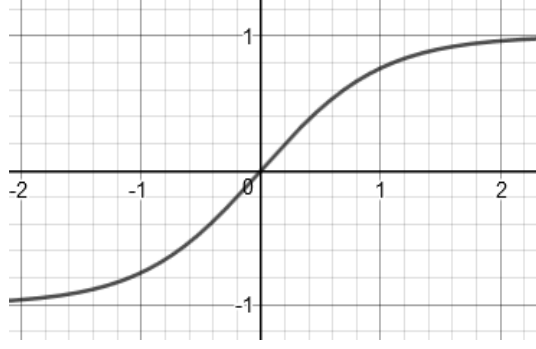
Şekil 3.5. Lojistik aktivasyon fonksiyonu

3.2.4.4. Tanjant Hiperbolik aktivasyon fonksiyonu

Tanjant Hiperbolik fonksiyonu da lojistik aktivasyon fonksiyonu gibi lineer olmayan bir aktivasyon fonksiyonudur. Fonksiyonun matematiksel ifadesi

$$f(net) = \frac{1 - e^{-2net}}{1 + e^{-2net}}, -\infty < net < \infty$$

şeklinde iken, grafiği Şekil 3.6’da gösterildiği gibidir.



Şekil 3.6. Tanjant Hiperbolik fonksiyonu

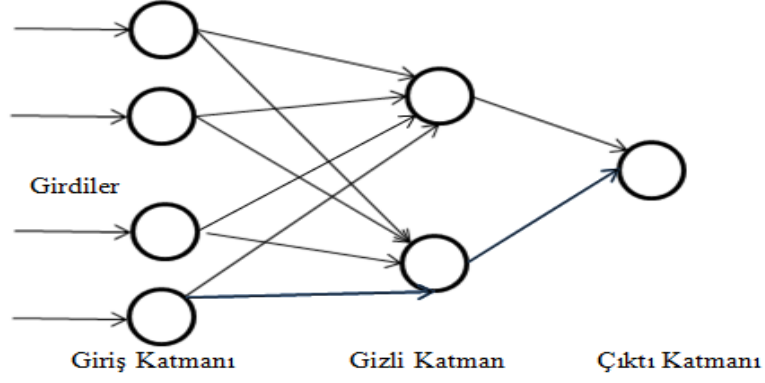
3.3. Yapay Sinir Ağı Mimarileri

YSA'lar, Bölüm 3.1'de verilen çok sayıda sinir hücresinin bir araya gelmesinden oluşur. YSA hücreleri ve bağlantıları çok değişik biçimlerde bir araya getirilebilir. Ağlar bu bağlantı şekillerine göre değişik mimari yapılarına sahip olurlar. YSA kullanılarak çözülen bir problemde mevcut çözümlerden daha iyi ya da optimal çözüm elde edilebilmesi için en iyi ağ mimarisinin oluşturulması gereklidir. Literatürde en çok kullanılan YSA mimarisi çok katmanlı algılayıcı (Multilayer Perceptron (MLP)) olup, bu mimari dışında Çarpımsal nöron model (Multiplicative Neuron), Pi-Sigma nöron modeli (Pi-Sigma Neuron Model) ve hibrit ağ mimarileri mevcuttur.

3.3.1. Çok katmanlı algılayıcı

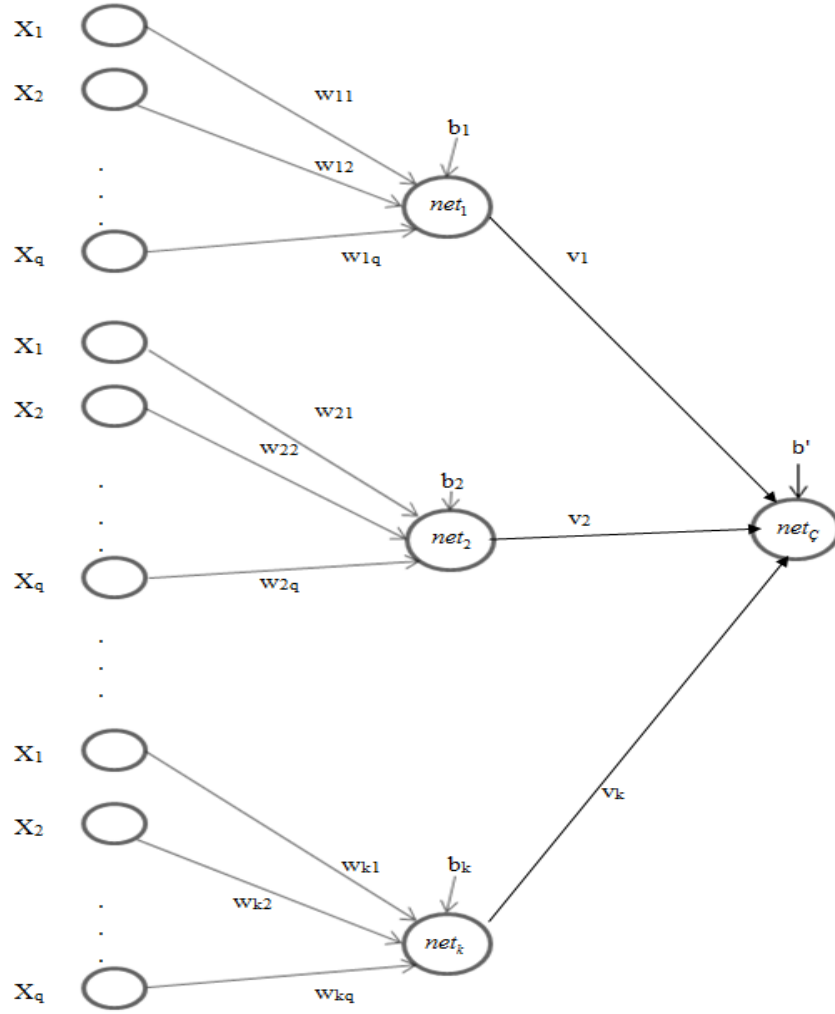
Çok sayıda sinir hücresinin bir araya gelmesi ile oluşan MLP mimarisi Şekil 3.7 ile gösterilmiştir. Aslında MLP, çok tabakalı lojistik regresyon modellerinden meydana gelmektedir (Bishop, 2006).

Bu tipte bir YSA'da girdi tabakası ile çıktı tabakası arasında en az bir adet gizli tabaka bulunur.



Şekil 3.7. Çok katmanlı YSA mimarisi

MLP mimarisi sınıflandırma problemlerinde kullanılacağından ve daha kolay anlaşılabilir olmasından dolayı bu mimariyi Şekil 3.8'deki gibi göstermek mümkündür. Şekil 3.2'de tanıtilen bir sinir hücresi dikkate alındığında Şekil 3.8'de birden fazla sinir hücresinin bir araya gelmesi ile MLP'nin oluştuğu ve gizli katmanında k adet hücre olduğu, saklı tabakadan çıkan değerlerin tekrar ağırlıklandırılarak çıktı hücresine iletiildiği ve son olarak burada aktivasyon fonksiyonu yardımıyla çıktı değerlerinin elde edildiği söylenebilir. Gizli katmandaki her bir hücre için toplama işlemi (3.2)'de verilmiştir. Toplama işlemi (3.3)'de verilen lojistik aktivasyon fonksiyonu ile tekrar ağırlıklandırılarak çıktı katmanındaki hücreye iletilir. Gizli tabakadaki hücrelerden çıkan değerler (3.4) hesaplanmasıyla çıktı hücresine gelir. Daha sonra tekrar aktivasyon fonksiyonundan geçerek çıktı değerleri (3.5) yardımıyla elde edilir.



Şekil 3.8. MLP mimarisinin farklı bir gösterimi

$$net_i = (x_1 w_{i1} + x_2 w_{i2} + \dots + x_q w_{iq}) + b_i, \quad i=1, \dots, k \quad (3.2)$$

$$u_i = f(net_i) = \frac{1}{1 + e^{-net_i}} \quad (3.3)$$

$$net_c = \sum_{i=1}^k u_i v_i + b' \quad (3.4)$$

$$\hat{y}_{(j)} = \frac{1}{1 + e^{-net_c}} \quad (3.5)$$

Burada;

x_i : Bağımsız değişken ya da girdi ($i=1, 2, \dots, q$)

q : Değişken sayısı

k : Gizli tabakadaki hücre sayısı

w_{ih} : i 'inci girdi değeri ile h 'inci gizli tabaka birimi arasındaki ağırlık değeri

b_i : Gizli tabakadaki i 'inci hücreye ait eşik değeri

net_i : Gizli tabakadaki i 'inci hücreye ait toplam değeri

u_i : Gizli tabakadaki i 'inci hücreye ait aktivasyon değeri

v_i : i 'inci gizli tabaka hücresi ile çıktı tabakası arasındaki ağırlık değeri

b' : Çıktı tabakasındaki hücreye ait eşik değeri

$net_ç$: Çıktı tabakasına ait toplam değeri

j : Veri kümesindeki gözlem numarası

$\hat{y}_{(j)}$: j . gözleme ait 1'inci gruba atanma olasılığıdır.

YSA'ların başarısı ağ yapısındaki ağırlıkların uygun değerleri almasına doğrudan bağlıdır. Bu yüzden başarılı çalışabilmeleri amacıyla YSA'lar eğitilirler; yani, verilen bir girdi vektör seti için hesaplanan çıktı değerleri ile hedeflenen çıktı değerlerinin birbirine yaklaşmasını sağlayacak optimum bağlantı ağırlıklarına ulaşmak hedeflenir. Nöronlar arası bağlantı ağırlıklarının ayarlanmasıyla, hesaplanan çıktı ile beklenen çıktı arasındaki fark yeterli bir duyarlılığa yaklaştığında YSA'nın öğrendiği sonucuna varılabilir. YSA'ların en iyi şekilde öğrenmesini sağlayacak ağırlık değerlerinin belirlenmesi yani YSA'nın eğitilmesi bilim insanlarının yıllardır çalıştığı optimizasyon problemlerinden bir tanesidir (Öztürk, 2011).

MLP mimarisine sahip bir YSA'da ağ eğitimi, Şekil 3.8'de yer alan parametrelerin en iyi değerlerinin elde edilmesi işlemidir. MLP'ye ait parametreler vektörü PV_{MLP} ile gösterilsin. Bu vektör $k \times (q + 2) + 1$ adet parametre içerir. Burada q girdi sayısı (bağımsız değişken sayısı), k gizli katmandaki hücre sayısıdır (Kolay vd 2016).

$$PV_{MLP} = [w_{11} \ w_{12} \ \dots \ w_{1q} \ b_1 \ | \ \dots \ | \ w_{k1} \ w_{k2} \ \dots \ w_{kq} \ b_k \ | \ v_1 \ \dots \ v_k \ b']_{k(q+2)+1} \quad (3.6)$$

MLP ağının eğitimi için ilk kez en küçük kareler algoritmasının genel bir hali olan geri yayılım algoritması önerilmiştir (Rumelhart vd 1986). Geri yayılım algoritması ile beklenen ve gerçek çıktı arasındaki fark eğilimli azalış (gradient descent) kullanılarak minimize edilir. Kareler toplamının en küçüklenmesi hatanın katman katman geriye doğru yayılması ile yapılır. Geri yayılım algoritması birçok

problem için YSA'yı başarılı bir şekilde eğitmesine rağmen yakınsama hızının yavaş olması bir sorun olarak düşünülmüştür. Bu nedenle Becker ve Cun (1988), geliştirilmiş geri yayılım algoritmasını önermişlerdir. Jacobs (1988) çalışmalarında, "Delta-Bar-Delta" öğrenme kuralı ile geri yayılım algoritmasının performansını artırmaya çalışmıştır. Geri yayılım algoritması ve benzeri türeve dayalı algoritmaların kullanımı dışında makine öğreniminin gelişmesi ve sezgisel algoritmaların etkinliği nedeni ile MLP eğitiminde bu algoritmalar da kullanılmıştır. Montana ve Davis (1990) yaptıkları çalışmada, MLP eğitimi için genetik algoritmayı kullanmayı önermişlerdir. Brent (1991), karar ağaçlarını kullanarak ve YSA'da saklı tabaka sayısına karar vermeye gerek kalmayacak şekilde etkili ve daha hızlı çalışan yeni bir algoritma önermiştir. Johansson vd (1992) çalışmalarında, birleşik eğimli azalış yöntemi (conjugate gradient descent) ile eğitim performansını yükseltmeyi amaçlamışlardır. Hagan ve Menhaj (1994) çalışmalarında, ağ eğitiminde "Levenberg Marquardt" algoritmasını kullanmışlardır. Parlos vd (1994) çalışmalarında, eğitim kuralı olarak en dik iniş (steepest descent) yöntemini kullanmış ve öğrenme oranını hatanın bir fonksiyonu olarak ele almışlardır. Salerno (1997) çalışmasında, ağ eğitimi için PSO'yu önermiştir. Zhang vd (2000) çalışmalarında, en iyi YSA yapısını bulmak için PSO'yu kullanmışlardır. Castillo vd (2000), genetik algoritma ile geri yayılım algoritmasını birlikte kullanarak MLP'nin eğitim performansını arttırmaya çalışmışlardır. Mendes vd (2002) çalışmalarında, sınıflandırma ve regresyon problemlerinin çözümü için ileri beslemeli çok katmanlı ağ kullanmışlar ve ağın eğitimi için PSO kullanmışlardır. Ilonen vd (2003) çalışmalarında, YSA eğitimi için difarensiyel evrim algoritmasını kullanmışlardır. Suratgar vd (2005) çalışmalarında, düzeltilmiş "Levenberg Marquardt" algoritması ile YSA eğitimini önermişlerdir. Ludermir vd (2006) çalışmalarında, tabu arama algoritması, benzetimli tavlama kullanarak yeni bir eğitim algoritması önermişlerdir. Zhang vd (2007) çalışmalarında, PSO ile geri yayılım algoritmasının hibrit bir yaklaşımını ağ eğitimi için önermişlerdir. Karaboğa vd (2007), ağ eğitimi için "Yapay Arı Kolonisi" optimizasyon algoritmasını önermişlerdir. Zanchettin vd (2011) yaptıkları çalışmada, sezgisel benzetimli tavlama, tabu arama algoritması, genetik algoritma ve geri yayılım algoritmasını birlikte kullanarak hem YSA mimarisinin hem de ağ parametrelerinin optimizasyonunu sağlamışlardır.

3.3.2. Pi-Sigma nöron model

Pi-Sigma nöron model ilk kez Shin ve Ghosh (1991) tarafından, fonksiyon yaklaşımı ve sınıflandırma problemleri çözümü için önerilmiştir. Yaptıkları çalışmada MLP ağının eğitim hızının yavaş olduğunu öne sürmüşler ve ara katmandan çıktı katmanına giden ağırlıkları sabit tutarak, çıktı değerlerinin ara tabakalardan etkilenmemesini sağlamışlardır. Ayrıca ağ eğitimi için eğilimli azalış yönteminin rastgeleleştirilmiş bir çeşidi olan olasılıksal öğrenme kuralını önermişlerdir. Pi-Sigma nöron modelinin mimarisi Şekil 3.9 ile gösterilmektedir. Verilen girdi seti için çıktı değerlerinin elde edilmesi (3.7), (3.8), (3.9) ve (3.10) eşitlikleri ile gösterilmiştir.

$$net_i = (x_1 w_{i1} + x_2 w_{i2} + \dots + x_q w_{iq}) + b_i, \quad i=1, \dots, k \quad (3.7)$$

$$u_i = f(net_i) = \frac{1}{1 + e^{-net_i}} \quad (3.8)$$

$$\Pi = \prod_{i=1}^k u_i \quad (3.9)$$

$$\hat{y}_{(j)} = 1 / (1 + e^{-\Pi_{(j)}}) \quad (3.10)$$

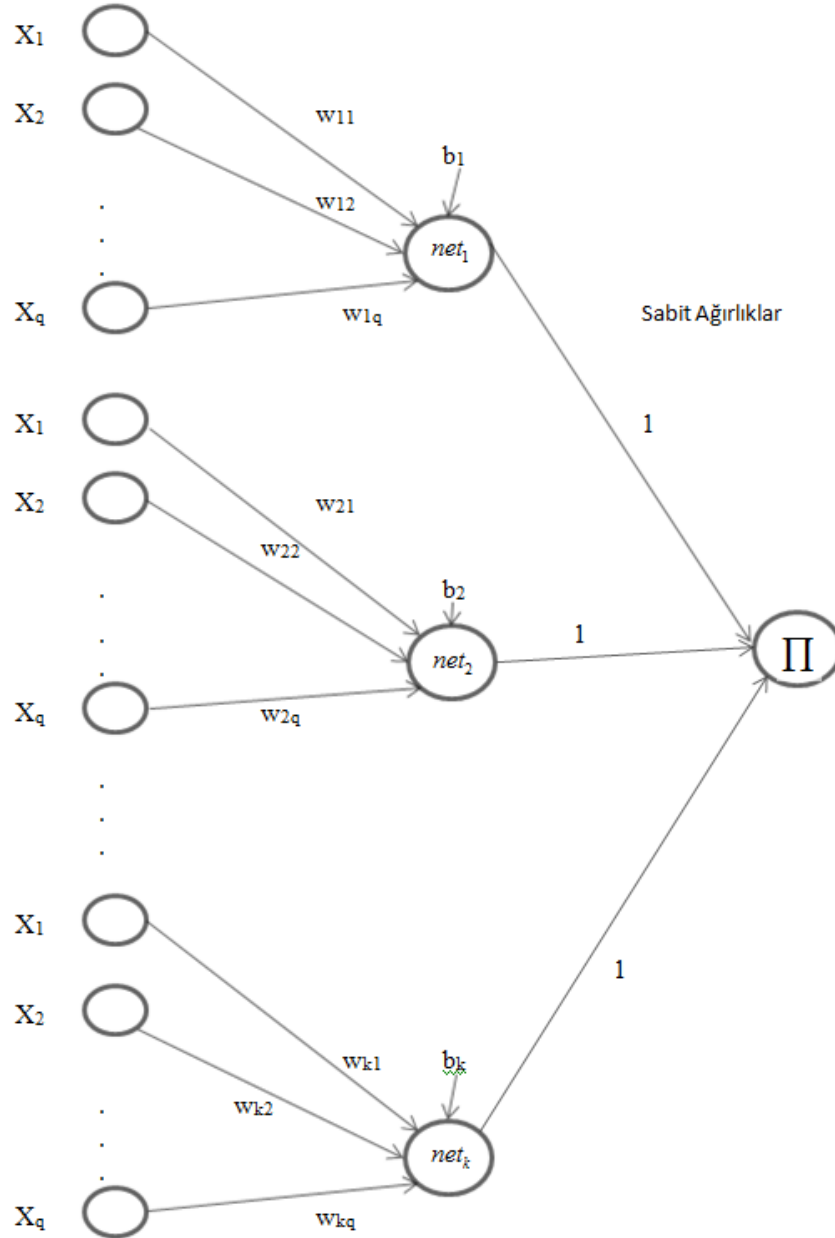
Pi-Sigma nöron modelde, MLP'nin aksine gizli tabaka yerine k adet toplam ünitesi vardır. Shin ve Ghosh (1991) çalışmalarında, sınıflandırma problemlerinin çözümünde k değerinin 3 ya da 4 olarak alınmasının uygulamada daha iyi sonuç verdiğini belirtmişlerdir. Hussain ve Liatsis (2002), tekrar eden Pi-Sigma nöron modeli önermiş ve bu modeli görüntü kodlama problemlerinde uygulamışlardır. Li (2003) yaptığı çalışmada, birden fazla Pi-Sigma ağını birleştirerek Sigma-Pi-Sigma modelini önermiştir. Ghazali ve Al-Jumeily (2009) yaptıkları çalışmada, Pi-Sigma ağını Ridge Polinomlu sinir ağı ile birleştirerek finansal zaman serisi problemlerine uygulamışlardır. Panigrahi vd (2013) yaptıkları çalışmada, Pi-Sigma sinir ağının eğitimi için diferansiyel evrim algoritmasını kullanmış ve sınıflandırma problemlerine başarı ile uygulamışlardır. Nayak vd (2014), genetik algoritma ile parçacık sürü optimizasyon algoritmasını birleştirerek Pi-Sigma ağının eğitiminde kullanarak sınıflandırma problemlerinde başarılı bir şekilde uygulamışlardır.

Pi-Sigma mimarisi için ağ eğitimi Şekil 3.9'da yer alan parametrelerin en iyi değerlerinin elde edilmesi işlemidir. Pi-Sigma ağına ait parametreler vektörü $PV_{Pi-Sigma}$ ile gösterilsin. Bu vektör $k(q+1)$ adet parametre içerir. Burada q girdi

sayısı (bağımsız değişken sayısı), k toplam ünitesi sayısıdır ve (3.11) ile verilen bu vektördeki parametrelerin değerlerinin belirlenmesi bir optimizasyon problemidir.

$$PV_{Pi-Sigma} = [w_{11} \ w_{12} \ \dots \ w_{1q} \ b_1 \ | \ \dots \ | \ w_{k1} \ w_{k2} \ \dots \ w_{kq} \ b_k]_{k(q+1)} \quad (3.11)$$

Şekil 3.9 incelendiğinde k toplam ünitesinden çıktıya bağlanan ağırlıkların sabit tutulduğu görülmektedir.



Şekil 3.9. Pi-Sigma nöron model mimarisi

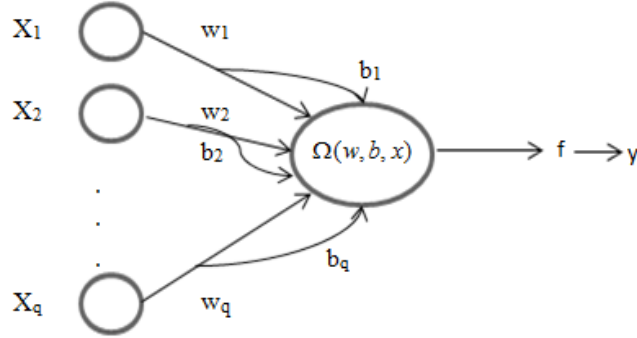
3.3.3. Çarpımsal nöron model

Çarpımsal nöron model ilk kez Yadav vd (2007) tarafından, zaman serisi kestirim problemlerinin çözümü için önerilmiştir. Çarpımsal nöron model, tek hücreli yapısından dolayı yüksek dereceli MLP ve Pi-Sigma modellerine göre çok daha basit yapıdadır. Ayrıca gizli katman olmaması çarpımsal nöron modelin avantajlı tarafı olarak düşünülebilir. Şekil 3.10 ile gösterilen Çarpımsal nöron modeli; bir hücreden oluşur ve MLP'den farklı olarak her bir girdi değişkeni için eşik değerlerinin eklenmesi ile oluşturulur. Çarpımsal nöron model için çıktı değerleri

$$\Omega = \prod_{i=1}^q (w_i x_i + b_i) \quad (3.12)$$

$$\hat{y}_{(j)} = \frac{1}{1 + e^{-\Omega(j)}} \quad (3.13)$$

eşitlikleri ile hesaplanır.



Şekil 3.10. Çarpımsal nöron model mimarisi

Burada q girdi sayısı ve j gözlem numarasıdır. Yadav vd (2007) yaptıkları çalışmada, çarpımsal nöron modelin zaman serisi kestiriminde MLP'den daha iyi sonuçlar verdiğini göstermişlerdir. Zhao ve Yang (2009) yaptıkları çalışmada, çarpımsal nöron modelde ağ eğitimi için uyarlanmış PSO kullanmış, önerdikleri yöntemi zaman serisi problemlerinde geri yayılım algoritması, standart PSO ve genetik algoritma ile karşılaştırmışlardır. Burse vd (2010) geri yayılım algoritmasında yerel optimuma takılma sorunu ile ilgilenmişler ve yerel optimuma takılma sorununu gidermek için geliştirilmiş geri yayılım algoritmasını önermişlerdir. Yolcu vd (2013), zaman serisi problemlerinin lineer ve lineer olmayan bileşenlere sahip olduğunu düşünmüşler ve bu tip problemlerin çözümü için hibrit bir

model önermişlerdir. Önerdikleri modelin lineer olmayan kısmında çarpımsal nöron modeli kullanmışlardır. Cui vd (2015) çalışmalarında, çarpımsal nöron modeline ileri yönlü ve geriye dönük eklentiler ekleyip, önerdikleri ağ eğitimi için evrimsel algoritmalarından ateş böceği sürü algoritmasını (glowworm swarm optimization) kullanmışlardır. Wu vd (2015) çalışmalarında, çarpımsal nöron modeli rüzgar hızı tahmininde kullanmışlardır. Gündoğdu vd (2015) çalışmalarında, çarpımsal nöron modelin eğitiminde lojistik aktivasyon fonksiyonu yerine Gauss aktivasyon fonksiyonunu kullanmayı önermişlerdir. Baş vd (2016) çalışmalarında, aykırı değer içeren gözlem kümeleri için Huber'in hata fonksiyonunu kullanarak dayanıklı bir eğitim algoritması önermişlerdir.

Çarpımsal nöron modelin eğitilmesi (3.14) ile verilen parametreler vektöründeki elemanların optimizasyonu ile yapılır. Bu vektör $2q$ adet ağ parametresinden oluşur.

$$PV_{CN} = [w_1 \ w_2 \ \dots \ w_q \ b_1 \ b_2 \ \dots \ b_q]_{2q} \quad (3.14)$$

3.4. Yapay Sinir Ağı Eğitimi

YSA'ların en önemli özelliği eğitilebilir olmasıdır. Bir YSA'nın eğitilmesi; verilen girdi değerleri ile modelleme sonucu elde edilen tahmini çıktı değerleri arasındaki farkı minimum yapacak ağırlık ve eşik değerlerinin elde edilmesi işlemidir. İlk kez Rumelhart vd (1986) tarafından MLP'yi eğitmek için önerilen geri yayılım algoritması, hatanın geriye doğru yayılması işlemidir. Geri yayılım algoritması türeve dayalı bir algoritmadır ve (3.15)'te verilen hata kareler toplamının türevi alınarak hatanın katman katman geri yayılması ile ağ parametrelerinin değerleri ayarlanır.

$$E = \frac{1}{2} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (3.15)$$

Gizli Tabaka ile Çıktı Tabaka Arası Ağırlık ve Eşik Değerlerinin Hesaplanması

Gizli tabaka ile çıktı tabakası arasındaki ağ parametrelerinin değerlerinin adimsal olarak nasıl değişmesi gerektiğine, (3.15) ile verilen hata fonksiyonunun (3.16) ve (3.17)'de verilen şekilde türevinin alınması ile karar verilir.

Burada;

y_i : i'inci gözlemin girdi değeri

\hat{y}_i : i'inci gözlem için elde edilen çıktı değeri

v_j : j'inci gizli tabaka ile çıktı tabakası arası ağırlık matrisi

net_j : j'inci gizli tabaka birimi için hesaplanan toplam değeri

net_{ζ} : Çıktı tabakası için hesaplanan toplam değer

u_j : j'inci gizli tabaka değeri için lojistik fonksiyon ile hesaplanan çıktı değeri

b' : Çıktı hücresine ait eşik değeri

η : Öğrenme oranıdır.

Çıktı tabakası ile gizli tabaka arasındaki ağırlık ve eşik değerleri sırasıyla (3.16) ve (3.17) ile hesaplanır. Hesaplamalar, türevde zincir kuralına göre yapılır.

$$v^{(t+1)} = v^{(t)} + \Delta v, \Delta v = -\eta \frac{\partial E}{\partial v} \quad (3.16)$$

$$b'^{(t+1)} = b'^{(t)} + \Delta b', \Delta b' = -\eta \frac{\partial E}{\partial b'} \quad (3.17)$$

Burada;

t: iterasyon numarası

$v^{(t)}$: t'inci iterasyondaki ağırlık matrisi

Δv : Ağırlık matrisindeki değişim değerleri

$v^{(t+1)}$: (t+1)'inci iterasyon için hesaplanan yeni ağırlık değerleri matrisi

$b'^{(t)}$: t'inci iterasyondaki çıktı tabakasına ait eşik değeri

$\Delta b'$: Eşik değerindeki değişim değeri

$b'^{(t+1)}$: (t+1)'inci iterasyon için hesaplanan yeni eşik değerini göstermektedir.

Ağırlık değerleri için türevde zincir kuralı uygulanarak

$$-\eta \frac{\partial E}{\partial v_j} = -\eta \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial net_{\zeta}} \frac{\partial net_{\zeta}}{\partial v_j} \quad (3.18)$$

eşitliği yazılabilir.

Eşitlik (3.18) ile verilen ifadede her bir türev işleminin sonucu aşağıdaki gibidir. Buna göre yeni ağırlık değerleri

$$\frac{\partial E}{\partial \hat{y}} = -(y - \hat{y}), \quad \frac{\partial \hat{y}}{\partial net_{\xi}} = \hat{y}(1 - \hat{y}), \quad \frac{\partial net_{\xi}}{\partial v_j} = u_j, \quad \text{olmak üzere;}$$

$$-\eta \frac{\partial E}{\partial v} = -\eta(-(y - \hat{y}))\hat{y}(1 - \hat{y})u_j \quad (3.19)$$

eşitliği ile hesaplanır. Eşik değeri için (3.17)'de türev alınarak (3.20) elde edilir. Yeni eşik değerleri (3.21) kullanılarak hesaplanır.

$$-\eta \frac{\partial E}{\partial b'} = -\eta \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial net_{\xi}} \frac{\partial net_{\xi}}{\partial b'} \quad (3.20)$$

$$-\eta \frac{\partial E}{\partial b'} = -\eta(-(y - \hat{y}))\hat{y}(1 - \hat{y}) \quad (3.21)$$

Girdi Tabakası ile Gizli Tabaka Arası Ağırlık ve Eşik Değerlerinin Ayarlanması

Girdi ile gizli tabakalar arasındaki parametre değerlerinin hesaplanması ise (3.22) ve (3.23) denklemlerinde gösterildiği gibi yapılır.

$$w^{(t+1)} = w^{(t)} + \Delta w, \quad \Delta w = -\eta \frac{\partial E}{\partial w} \quad (3.22)$$

$$b^{(t+1)} = b^{(t)} + \Delta b, \quad \Delta b = -\eta \frac{\partial E}{\partial b} \quad (3.23)$$

Burada;

w: Girdi ile gizli tabaka arası ağırlıklar matrisi

b: Girdi ile gizli tabaka arası eşik değerleri

x: Girdi Matrisi değerleri

$w^{(t)}$: t'inci iterasyondaki ağırlık değeri

$w^{(t+1)}$: (t+1)'inci iterasyondaki ağırlık değeri

Δw : t'inci iterasyon ile (t+1)'inci iterasyon arasındaki ağırlık değerindeki değişim miktarı

$b^{(t)}$: t'inci iterasyondaki eşik değeri

$b^{(t+1)}$: (t+1)'inci iterasyondaki eşik değeri

Δb : t'inci iterasyon ile (t+1)'inci iterasyon arasındaki eşik değerindeki değişim miktarıdır.

Girdi tabakası ile gizli katman arasındaki ağırlık ve eşik değerlerinin elde edilmesinde yine zincir kuralı kullanılarak sırasıyla (3.24) ve (3.25) elde edilir.

$$-\eta \frac{\partial E}{\partial w_{ij}} = -\eta \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial net_{\xi}} \frac{\partial net_{\xi}}{\partial u_j} \frac{\partial u_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ij}} \quad (3.24)$$

$$-\eta \frac{\partial E}{\partial b} = -\eta \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial net_{\xi}} \frac{\partial net_{\xi}}{\partial u_j} \frac{\partial u_j}{\partial net_j} \frac{\partial net_j}{\partial b} \quad (3.25)$$

Ağırlık değerlerinin elde edilmesi için (3.24) ile verilen ifadede türevler alındığında

$$\frac{\partial net_{\xi}}{\partial u_j} = v_j, \quad \frac{\partial u_j}{\partial net_j} = u_j(1-u_j), \quad \frac{\partial net_j}{\partial w_{ij}} = x_i$$

elde edilir. Bu sonuçlara göre yeni ağırlık değerleri (3.26) aracılığıyla hesaplanır.

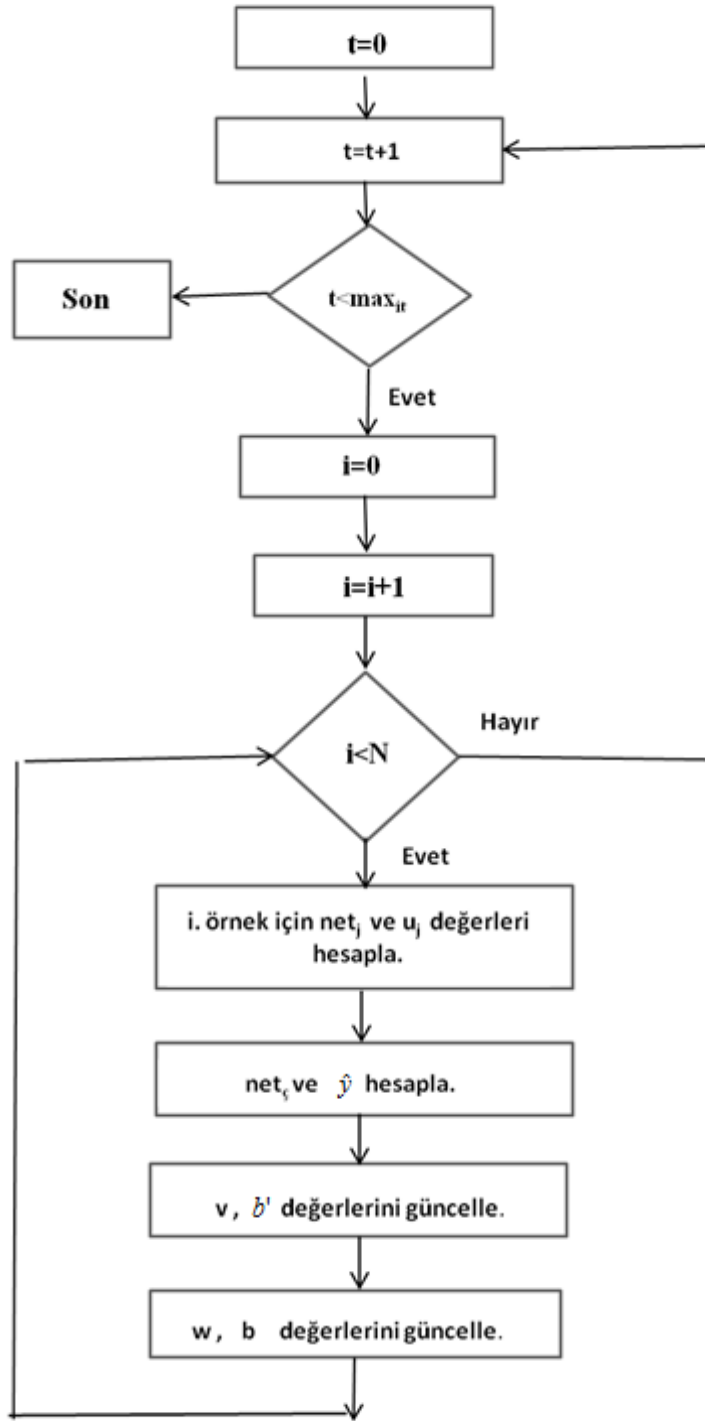
$$-\eta \frac{\partial E}{\partial w_{ij}} = -\eta(-(y-\hat{y}))\hat{y}(1-\hat{y})v_j u_j(1-u_j)x_i \quad (3.26)$$

Son olarak girdi tabakası ile gizli tabaka arasındaki eşik değerlerinin elde edilmesi için (3.25) verilen ifadede türevler hesaplanacak olursa (3.27) elde edilir.

$$-\eta \frac{\partial E}{\partial b} = -\eta(-(y-\hat{y}))\hat{y}(1-\hat{y})v_j u_j(1-u_j) \quad (3.27)$$

Çarpımsal nöron model ve Pi-Sigma sinir ağı modeli içinde, hesaplamalar hata fonksiyonlarının türevlenip hatanın katman katman geri yayılması ile yapılabilir.

Şekil 3.11'de geri yayılım algoritmasına ait akış şeması verilmiştir.



Şekil 3.11. Geri yayılım algoritması

4. PARÇACIK SÜRÜ OPTİMİZASYONU

Parçacık sürü optimizasyonu (PSO), ilk kez Eberhart ve Kennedy (1995) tarafından önerilmiş evrimsel optimizasyon algoritmasıdır. PSO; doğadaki kuş, balık ve benzeri canlı sürülerinin sosyal davranışlarının bir benzetimi olarak düşünülmüştür. PSO'da her bir elemana parçacık denir ve parçacıkların tümü sürüyü oluşturur. Her parçacık başlangıçta rastgele konum ve hız vektörüne sahiptir. D boyutlu arama uzayında i. parçacığa ait konum ve hız vektörü sırasıyla $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ ve $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ olmak üzere PSO'da her bir parçacık kendi geçmiş en iyi deneyimini ve tüm sürüye ait geçmiş en iyi deneyimi kullanarak hızını ve konumunu (4.1) ve (4.2)'ye göre günceller. Her bir parçacık arama uzayında en iyi çözüm için bir adaydır.

$$V_{ij}(t+1) = w * V_{ij}(t) + C_1 * rand * (P_i(t) - X_{ij}(t)) + C_2 * rand * (P_g(t) - X_{ij}(t)) \quad (4.1)$$

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1) \quad (4.2)$$

Burada;

t: İterasyon numarası

i: Parçacık numarası

j: Boyut indisi (j=1,2,...,D)

w: Başlangıç ağırlığı,

C_1 ve C_2 : Sosyalleşme parametreleri,

rand: (0,1) aralığında rastgele bir sayı,

P_i : i. parçacığın geçmiş en iyi deneyimi,

P_g : Tüm parçacıkların yani sürünün geçmiş en iyi deneyimi

$V_{ij}(t)$: j'inci boyuttaki i'inci parçacığın t'inci iterasyondaki hızı

$V_{ij}(t+1)$: j'inci boyuttaki i'inci parçacığın (t+1)'inci iterasyondaki hızı

$X_{ij}(t)$: j'inci boyuttaki i'inci parçacığın t'inci iterasyondaki konumu

$X_{ij}(t+1)$: j'inci boyuttaki i'inci parçacığın (t+1)'inci iterasyondaki konumu

Orijinal PSO'da başlangıç ağırlığı mevcut değildir. Shi ve Eberhart (1998), yaptıkları çalışmada başlangıç ağırlığını yeni bir parametre olarak önermişlerdir. Literatürde PSO, başlangıç ağırlığının da dâhil olduğu (4.1) ve (4.2)'de verilen şekli ile kullanılmaktadır.

PSO, Bölüm 3.2'de verilen farklı YSA mimarilerinin eğitimi için kullanılmıştır. MLP, Pi-Sigma ve Çarpımsal nöron model ağlarının PSO ile eğitilmesi sırasıyla (3.6), (3.11) ve (3.14) ile verilen eşitliklerdeki parametre vektörlerinin optimizasyon işlemidir.

Tüm evrimsel algoritmalar gibi PSO da yerel optimuma takılma sorununa sahiptir. Bu nedenle PSO'nun performansının geliştirilmesi üzerine birçok çalışma yapılmıştır. Clerc (1999), PSO'nun yakınsamasının garanti olması için kısıtlayıcı faktörlere ihtiyaç olduğunu belirtmiş ve yeni bir algoritma önermiştir. Eberhart ve Shi (2000) çalışmalarında, başlangıç ağırlığı değerlerinin 0.9 ile 0.4 arasında lineer azalan alınmasının, PSO'nun yakınsama performansını artıracaklarını önermiş ve aynı zamanda başlangıç ağırlığını C_1 ve C_2 parametrelerinin bir fonksiyonu olarak alıp karşılaştırma yapmışlardır. Bergh ve Engelbrecht (2001) çalışmalarında, girdi vektörünün alt vektörlere bölünerek, her vektörün kendi alt sürüsüne atanması önerilmiştir. Önerdikleri ve işbirlikçi (cooperative) eğitim algoritması dedikleri algoritmayı YSA'ların eğitiminde kullanmışlardır. Shi ve Eberhart (2001), bulanık sistemleri PSO'da başlangıç ağırlığının seçimine uygulamışlar ve üç farklı karşılaştırma fonksiyonu ile önerdikleri yöntemi test etmişlerdir. Juang (2004), genetik algoritma ile PSO'yu birleştirerek YSA üzerinde uygulamıştır. Mendes vd (2004), yaptıkları çalışmada her parçacığın diğer tüm komşularının bilgisinden yararlandığı bütünüyle bilgi içeren PSO önermişlerdir. Da ve Xiurun (2005), benzetimli tavlama tekniğini (Simulated Annealing) PSO'ya uyarlamışlardır. Liang vd (2006), kapsamlı öğrenme kuralı ile PSO'yu yeniden düzenlemiştir. Huang ve Mohan (2006), yaptıkları çalışmada, mikro PSO'yu önermişler ve çok boyutlu problemlerin çözümünde kullanmışlardır. Malik vd (2007), başlangıç ağırlığını lojistik artan fonksiyon kullanarak ayarlamışlardır. Feng vd (2007), PSO için kaotik azalan başlangıç ağırlığını önermişlerdir. Xin vd (2009) çalışmalarında, çoklu evreli lineer azalan başlangıç ağırlığını önererek PSO'nun performansını arttırmaya çalışmışlardır. Lin ve Chen (2009) çalışmalarında, diskriminant analizinde değişken seçimi için PSO'yu kullanmıştır. Liu vd (2011) çalışmasında, tüm sürüyü

alt sürülere bölerek yeni bir algoritma önermiş ve sınıflandırma problemleri için değişken seçiminde kullanmışlardır. Bansal vd (2011) yaptıkları çalışmada, 15 farklı başlangıç stratejisini ele almış, bu yöntemleri beş farklı karşılaştırma fonksiyonu kullanarak karşılaştırmıştır. Yolcu vd (2013) yaptıkları çalışmada, başlangıç ağırlığı ve kısıtlayıcı faktörlerin değerlerini adimsal olarak değiştirerek uyarlanmış PSO'yu zaman serisi kestirim problemleri için önermişlerdir. Geng vd (2013) çalışmalarında, başlangıç ağırlığı değerinde yeni bir düzenleme yapmış ve başlangıç değeri ile kısıtlayıcı faktör değerlerini optimizasyon sürecinde adimsal olarak değiştirmeyi önermişlerdir. Chan ve Chen (2015) yaptıkları çalışmada, 0 ile 1 arasında değer alan rastgele değişkenin yüksek değerler alması durumunda yerel optimuma takılma sorunu ile karşılaşılabilceğini bu nedenle 0.7'den büyük rastgele değerler için tekrardan atanma yapılmasını önermişlerdir. Liang vd (2015) çalışmalarında, kümelemeye dayalı uyarlanmış PSO'yu önermişler ve karşılaştırma fonksiyonları kullanarak yöntemlerini diğer PSO çeşitleri ile karşılaştırmışlardır.

4.1. Uyarlanmış Parçacık Sürü Optimizasyonu (UPSO)

PSO, kolay hesaplanabilirliği ve diğer evrimsel algoritmalara göre daha kolay anlaşılabilir olmasından dolayı YSA eğitiminde sıkça kullanılmaktadır. Etkin çözüm yeteneğine sahip olmasına rağmen, diğer tüm evrimsel algoritmalar gibi PSO da yerel optimuma takılabilir ve yakınsama hızı yavaşlayabilir. UPSO ile PSO'nun performansını artırmak için w , C_1 ve C_2 parametreleri iterasyona bağlı olarak değiştirilmiştir. Önerilen modifikasyonların standart PSO'ya göre daha iyi performans gösterip göstermediği Bölüm 7.1'de matematikte sıklıkla kullanılan karşılaştırma fonksiyonları üzerinde test edilmiştir.

$$w(t) = (w_{\text{maks}} - w_{\text{min}}) \left(\frac{t_{\text{maks}} - t}{t_{\text{maks}}} \right) + w_{\text{min}} \quad (4.3)$$

$$c_1(t) = (c_{1\text{maks}} - c_{1\text{min}}) \left(\frac{t}{t_{\text{maks}}} \right) + c_{1\text{min}} \quad (4.4)$$

$$c_2(t) = (c_{2\text{maks}} - c_{2\text{min}}) \left(\frac{t}{t_{\text{maks}}} \right) + c_{2\text{min}} \quad (4.5)$$

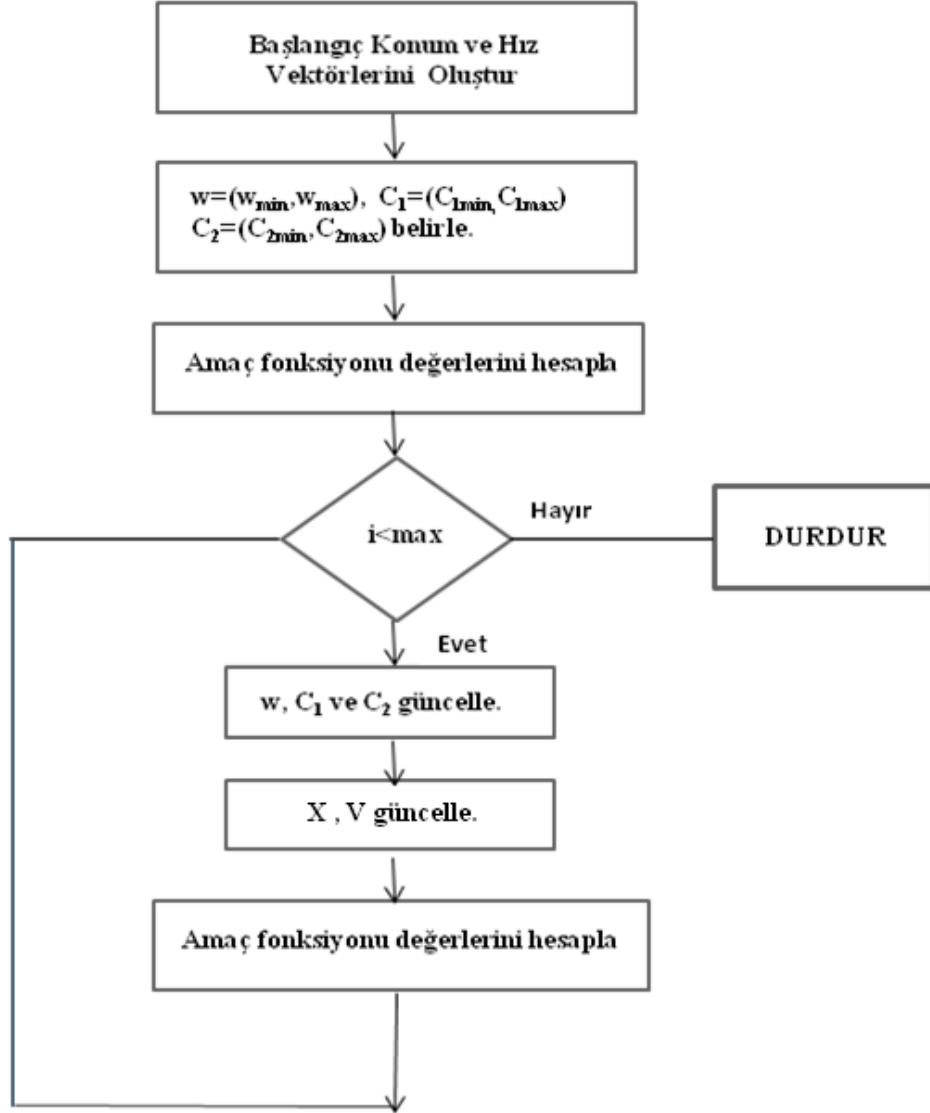
Burada;

$(w_{\text{min}} - w_{\text{maks}})$: Başlangıç ağırlık değeri w 'ye ait aralığın sırasıyla alt ve üst değerleri

$(c_{1min} - c_{1maks})$: C_1 parametresine ait aralığın sırasıyla alt ve üst değerleri

$(c_{2min} - c_{2maks})$: C_2 parametresine ait aralığın sırasıyla alt ve üst değerleri

t_{maks} : Maksimum iterasyon sayıdır.



Şekil 4.1. UPSO algoritması

5. LOJİSTİK REGRESYON ve ÇARPIMSAL NÖRON MODELE DAYALI YENİ BİR SINIFLANDIRICI

Sınıflandırma problemlerinin çözümü için literatürde birçok sınıflandırıcı önerilmiştir. Bunlardan bazıları Diskriminant Analizi, Bayesci Sınıflandırma, Destek Vektör Makineleri, Karar Ağaçları, Kural Tabanlı Sınıflandırıcılar, Genelleştirilmiş Lineer Modeller, En Yakın Komşuluk, Kısmi En Küçük Kareler ve Temel Bileşen Regresyon, Lojistik ve Çok Terimli Regresyon'dur.

Wolpert (1996) "No-Free-Lunch" teoremine göre en iyi sınıflandırıcı tüm veri kümeleri için aynı olmayacaktır. Başka bir deyişle, herhangi bir sınıflandırma problemi için önerilen bir sınıflandırıcı tüm veri kümeleri için diğer yöntemlerden daha iyi olamaz. Bu nedenle farklı sınıflandırıcılar birlikte kullanılarak hibrit modeller önerilmiştir.

Bölüm 2'de lojistik regresyon yöntemi tanıtılıp, lojistik regresyonda parametre tahminlerinin en çok olabilirlik yöntemi ile elde edilmesinden bahsedilmişti. En çok olabilirlik yöntemi ile yüksek eğitim performansının elde edilmesi, aşırı uyum (overfitting) problemi nedeni ile gelecekteki yeni bir gözlemin hangi sınıfta yer alacağına dair iyi bir gösterge olamaz (Bishop, 2006). Ayrıca LR, modele bağlı bir yaklaşımken, YSA veriye bağlı bir yaklaşımdır (Tunc, 2012). Veri kümesinde daha önce hiç yer almamış gözlemlerin sınıflandırılmasında yani test kümesinde yüksek doğru sınıflandırma performansının elde edilmesinde, aktivasyon fonksiyonu olarak lojistik aktivasyon fonksiyonu seçilerek birden fazla lojistik modelin kullanılması ile MLP'nin oluşturulduğundan bahsedilmişti. MLP'de yüksek test performansı elde edilmesine rağmen yakınsama hızının yavaş olması ve parametre sayısının fazla olması bir sorun olarak görülmüş ve bu sinir ağına alternatif sinir ağları, Pi-Sigma ve Çarpımsal Nöron model tanıtılmıştı. Ayrıca ilk kez MLP'nin eğitimi için önerilen geri yayılım algoritmasının yavaş çalışması, hesaplama zorluğu gibi problemler nedeniyle bu algoritma yerine evrimsel algoritmalarından olan PSO algoritması tanıtılmıştı. Bu bölümde ise lojistik regresyon ve çarpımsal nöron modele dayalı yeni bir sinir ağı modeli önerilmektedir.

Şekil 5.1 ile verilen modelde, k adet lojistik regresyon modeli girdi olarak düşünülmüş ve bu modeller çarpımsal nöron modelin girdi tabakasını oluşturmuştur. Önerilen modelde çıktı değerinin hesaplanması (5.1) ve (5.2) yardımıyla yapılır.

$$\Sigma_i = \sum_{j=1}^q w_{ij} x_j + b_{i1} ; i=1,2,\dots,k \quad (5.1)$$

$$\Pi = \prod_{i=1}^k \left(\frac{1}{1 + e^{-\Sigma_i}} v_i + b_{i2} \right) \quad (5.2)$$

Burada;

q : Girdi sayısı

k : Kullanılacak lojistik model sayısı

w_{ij} : i 'inci lojistik modelin j 'inci değişkenine ait ağırlık değeri

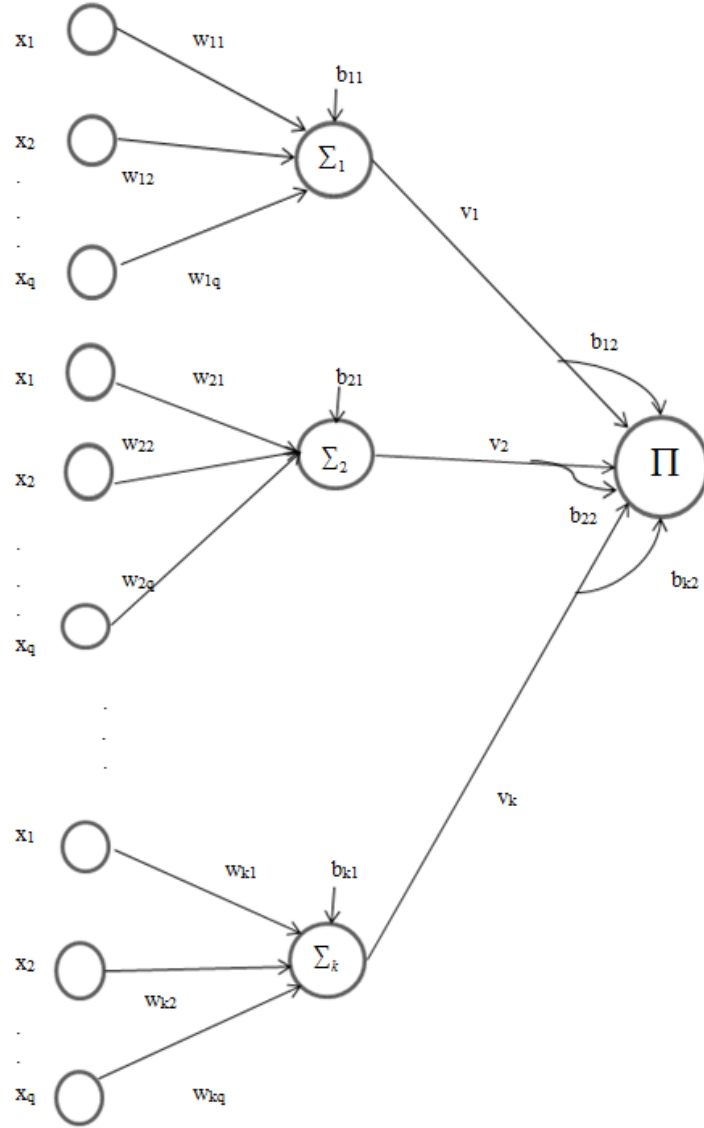
b_{i1} : i 'inci lojistik modelin eşik değeri

Σ_i : i 'inci lojistik modele ait toplam değeri

v_i : i 'inci lojistik modeli çıktı ünitesine bağlayan ağırlık değeri

b_{i2} : Çıktı birimlerine ait i 'inci lojistik modellerden gelen eşik değerleridir.

Önerilen bu ağ mimarisi bir bütün olarak ele alındığında, SMN ağ mimarisi olarak düşünülebilir. Bu modelde girdi tabakasını oluşturan sinir hücrelerinin lojistik aktivasyon fonksiyonu kullanılarak aktive edilmesi, lojistik modelin kurulabilmesi açısından zorunludur. Önerilen yeni YSA mimarisinde k sayısının alacağı değer hakkında herhangi bir kısıt yoktur. Ancak; MLP ve Pi-Sigma nöron modellerinde olduğu gibi bu değer de deneme yanılma yöntemi ile belirlenebilir. Ek olarak k değerinin mümkün olduğunca küçük seçilmesi, daha az sayıda eğitim parametresi kullanılmasına bu nedenle de ağ eğitimi için harcanacak sürenin kısılmasına olanak sağlayabilir.



Şekil 5.1. Lojistik regresyon ve çarpımsal nöron modele dayalı yeni sınıflandırıcı mimarisi

Şekil 5.1 ile gösterilen hibrit modelin ağ eğitimi (5.3) ile gösterilen parametreler vektörünün optimizasyonu ile gerçekleşir. Bu vektör Bölüm 4.'te tanımlanan PSO için kullanılacak ve tüm ağ parametreleri PSO'daki parçacıkların konumuna karşılık gelecek şekilde değerlendirilecektir.

$$PV_{Hibrit} = [w_{11} \ w_{12} \dots w_{1q} \ b_{11} \ ; \dots \ ; w_{k1} \ w_{k2} \ \dots w_{kq} \ b_{k1} \ ; v_1 \ v_2 \dots v_q \ ; b_{12} \ b_{22} \dots b_{k2}]_{k(q+3)} \quad (5.3)$$

Önerilen bu hibrit yaklaşımın yapısı Pi-Sigma nöron model ile de benzerlik göstermektedir. Ancak burada k adet toplam ünitesi ile çıktı tabakası arası sabit

ağırlıklarla bağlanmamıştır. Böylece lojistik modellerde oluşabilecek herhangi bir hatada, ağ çıktı değerinin etkilenme oranı azalabilmektedir. Bölüm 3.2’de verilen MLP, Pi-Sigma ve Çarpımsal Nöron model ağ mimarileri içinde optimize edilecek parametre sayısı en fazla olan ağ yapısı MLP’dir. MLP’de gizli katman tabaka sayısının belirlenmesi üzerine herhangi bir kısıt olmamasına rağmen genelde gizli tabaka sayısı 8-10 arası bir değer alınmaktadır.

Özellikle fazla sayıda mimari unsurları, tabaka sayısı ve tabakalarda fazla sayıda hücrelerin bulunması durumunda, YSA matematiğinin eğitilmesinde eğitim verileri yetersiz kalabilir. Bu bakımdan genel bir kural olarak mimari unsurları daha az olan YSA mimarileri tercih edilir (Şen, 2004).

Önerilen bu modelde ise toplam girdi tabakası eleman sayısının (kullanılacak lojistik model sayısının) 2, 3 ya da 4 alınmasının yeterli olduğu görülmüştür. Bu nedenle de önerilen modelin parametre sayısı MLP’den daha az ancak diğer iki yöntemden daha fazla olabilmektedir.

6. ÇAPRAZ GEÇERLİLİK

Çapraz Geçerlilik (Cross Validation), en çok kullanılan model ya da yöntem seçim ölçütlerinden biridir. Temel olarak veri kümesinin bölünmesi ile bir kısmının model oluşturulması ve kalan kısmının ise modelin performansının ölçülmesinde kullanılır. Performansı hesaplanan modeller içinde toplam performansı en yüksek olan modelin seçilme işlemidir (Yang, 2007).

En bilindik çapraz geçerlilik yöntemleri; direkt bölme çapraz geçerlilik (Hold-Out cross validation) , K-kat bölme çapraz geçerlilik (K-fold cross validation) ve birini dışarıda bırak çapraz geçerlilik (Leave-One-Out cross validation) yöntemleridir. (Browne, 2000; Efron 1983).

Şekil 6.1’de gösterildiği gibi direkt bölme çapraz geçerlilik ile veri eğitim kümesi ve geçerlilik kümelerine rastgele bölünür. Genelde verinin %70’i eğitim kalan %30’u ise geçerlilik kümesini oluşturur. Ancak; bu oran problem tipine göre farklılıklar gösterebilmektedir. Bu yöntem ile eğitim kümesi ve geçerlilik kümesinde aynı veri yer almaz. Direkt bölme çapraz geçerlilik için hata oranı

$$E_{Hold-out} = \frac{\text{Geçerlilik kümesindeki hatalı sınıflandırma sayısı}}{\text{Geçerlilik kümesindeki veri sayısı}} \quad (6.1)$$

eşitliği ile hesaplanır.

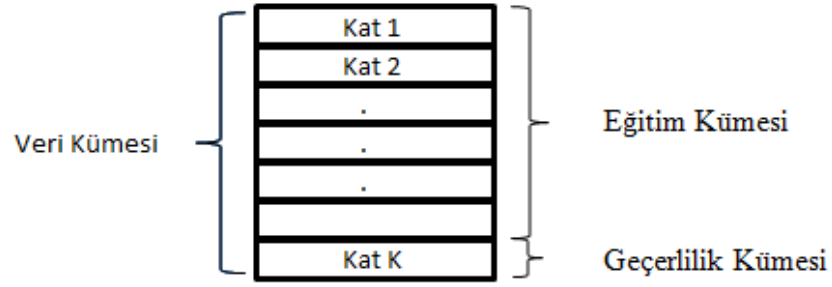


Şekil 6.1. Direkt bölme çapraz geçerlilik yöntemi

Direkt bölme çapraz geçerlilik yönteminde modelin geçerliliği için tüm veri kullanılmaz ve hata oranı verinin nasıl bölündüğüne yüksek derecede bağlıdır.

K-kat çapraz geçerlilik ile bu tip bir bağımlılık ortadan kaldırılabılır (Kohavi, 1995). Şekil 6.2’de gösterildiği gibi K-kat çapraz geçerlilikte veri kümesi eşit sayıda k büyüklükte alt gruplara bölünür. Adımsal süreçte her bir alt küme ($k-1$) kez eğitim kümesi için, bir kez de geçerlilik kümesi için kullanılır. K-kat çapraz geçerlilikte tüm veriler geçerlilik kümesinde kullanıldığından, sınıflandırma hatası oranı iyileştirilir. Ancak k değerinin artması hatalı sınıflandırma oranı tahmininin varyansını ve hesaplama zamanını artırır.

Birini dışarıda bırak çapraz geçerlilik, K-kat çapraz geçerliliğin özel bir ($k=N$) durumudur. Veri, gözlem sayısı kadar alt gruba bölünmüş olur. Her bir gözlem N defa eğitim kümesinde, 1 defa da geçerlilik kümesinde yer alır. Birini dışarıda bırak çapraz geçerlilik ise hatalı sınıflandırma oranının yansız bir tahminidir ancak büyük varyansa sahiptir.



Şekil 6.2. K-kat çapraz geçerlilik

Model seçimi için hangi çapraz geçerlilik yönteminin seçileceği veri kümesinin büyüklüğüne bağlıdır. Eğer veri kümesi küçükse birini dışarıda bırak çapraz geçerlilik yansız tahmin verir. Ancak veri kümesi yeterince büyükse K-kat çapraz geçerlilik hem hesaplama zamanını azaltır hem de tahminin doğruluğunu iyileştirebilir. K değerinin seçimi örneğe göre değişebilir ancak literatürde en çok 10 kat çapraz geçerlilik kullanılmaktadır (Cheng ve Pecht, 2012).

K-kat çapraz geçerlilik için her bir geçerlilik kümesi için hata oranı (6.2) ile bulunur. Daha sonra hata oranı K-kat için (6.3) ile verildiği şekilde hesaplanır.

$$E_i = \frac{\text{i'inci İterasyondaki Hatalı Sınıflandırma Sayısı}}{\text{i'inci İterasyonda Geçerleme Kümesindeki Veri Sayısı}} \quad (6.2)$$

$$E_{K-fold} = \frac{1}{K} \sum_{i=1}^K E_i \quad (6.3)$$

Birini dışarıda bırak çapraz geçerlilik K-kat çapraz geçerliliğin $k=N$ olduğu durumdur. Gözlem sayısı kadar hata oranı hesaplanır ve (6.4) ile verildiği gibi bu oranların ortalaması hesaplanarak toplam hata oranı elde edilir.

$$E_{Leave-one-out} = \frac{1}{N} \sum_{i=1}^N \text{Her İterasyondaki Hatalı Sınıflandırma Sayısı} \quad (6.4)$$

Kohavi (1995) çalışmasında, yaptığı 500 bin deneme ile model seçimi için 10 kat çapraz geçerliliğin diğer çapraz geçerlilik yöntemlerinden daha iyi olduğunu göstermiştir. Bölüm 6.2 ile verilen sınıflandırma problemlerinin çözümünde hangi problem için hangi yöntemin seçileceğine veri kümeleri yeterince büyük olduğundan 10-kat çapraz geçerlilik ile karar verilecektir. Bu problemlerin çözümünde 10-kat çapraz geçerlilik ile veri kümesi rastgele olarak 10 alt kümeye bölünür. İteratif süreçte her bir alt küme 1 kez geçerlilik kümesi olarak ele alınır.

Karşılaştırılan sınıflandırıcıların ortalama kestirim hatası bulunarak, bu sınıflandırıcıların veri kümesi tarafından daha önce hiç görülmemiş gelecek verileri doğru sınıflandırma performansları karşılaştırılarak hangi sınıflandırma problemi için hangi sınıflandırıcının daha iyi olduğuna karar verilmiştir.

7. BULGULAR ve TARTIŞMA

Sınıflandırma problemlerinin çözümü için YSA kullanıldığında doğru sınıflandırma performansının artırılması için eğitim algoritmasında, ağ yapısında ya da her ikisinde de modifikasyon yapılabilir.

Bu tez çalışmasında, Bölüm 4.1 ile verilen UPSO ile farklı YSA modellerine ait eğitim performansları geliştirilmeye ve Bölüm 5'te önerilen yeni sınıflandırıcı ile sınıflandırma problemlerinin çözümünde daha iyi performans elde edilmeye çalışılmıştır.

Tezin bu bölümünde ise öncelikle Bölüm 7.1 ile UPSO'nun PSO'ya göre daha iyi performans gösterip göstermediği deneysel olarak araştırılmıştır. Daha sonra Bölüm 7.2'de direkt bölme çapraz geçerlilik yöntemi ile Bölüm 3'te tanıtılan sinir ağları ve Bölüm 5'te önerilen yeni sinir ağı sınıflandırıcılarının performansları incelenmiştir. Son olarak Bölüm 7.3 ile Çizelge 7.3'te verilen gerçek hayat problemlerinin çözümünde hangi sınıflandırıcının daha iyi olduğu araştırılmıştır.

7.1. PSO ve UPSO Algoritmalarının Karşılaştırılması

UPSO algoritmasının standart PSO'dan daha iyi performans gösterip göstermediği, Çizelge 7.1 ile verilen ve Xu vd (2015) çalışmasında, kullanılan 8 adet karşılaştırma fonksiyonu, 30 boyutlu olarak ele alınmış ve 100 bağımsız deneme ile araştırılmıştır. Her iki algoritma için 1000 iterasyon ve sürü büyüklükleri 30 olarak alınmıştır. Başlangıç ağırlık değerleri, Standart PSO için 0.9 ile 0.4 arasında lineer azalan olarak, UPSO için ise yine 0.9 ile 0.4 arasında (4.3) eşitliğinde verildiği gibi alınmıştır. Kısıtlayıcı faktörler ise standart PSO için $c_1=c_2=1.4995$ olarak ele alınmış, UPSO için ise bu değerler (1.4995-1.2) aralığında (4.4) ve (4.5) ile verilen eşitlerdeki gibi hesaplanmıştır. Ayrıca bu fonksiyonların grafikleri ve iki algoritma kullanılarak elde edilen çözüme yakınsama hızlarının grafikleri Şekil 7.1 - Şekil 7.8 ile verilmiştir. Çizelge 7.1 ile verilen fonksiyonların çözümü için PSO ve UPSO kullanılarak elde edilen sonuçlar Çizelge 7.2'de yer almaktadır.

Çizelge 7.1. Karşılaştırma fonksiyonları

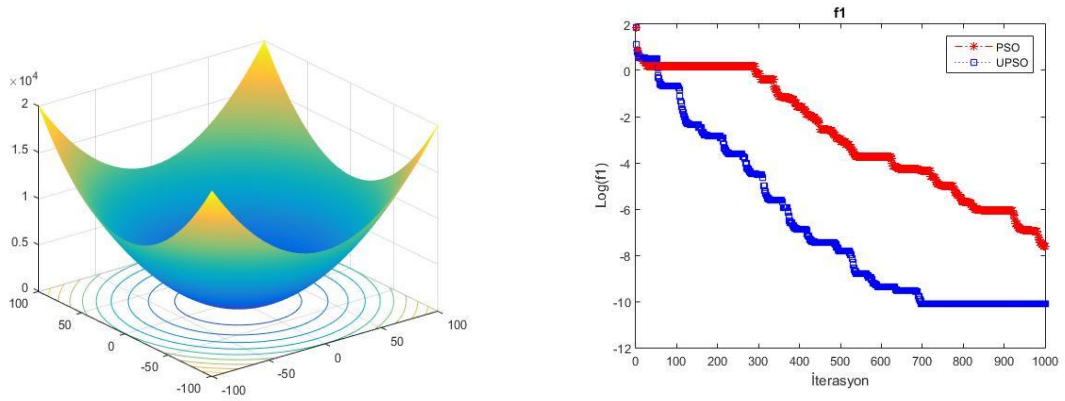
| Fonksiyon | Matematiksel Gösterim | Arama Uzayı | Optimum |
|-----------|---|-------------------|---------|
| $f_1(x)$ | $f_1(x) = \sum_{i=1}^D x_i^2$ | $[-100, 100]^D$ | 0 |
| $f_2(x)$ | $f_2(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ | $[-600, 600]^D$ | 0 |
| $f_3(x)$ | $f_3(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$ | $[-5.12, 5.12]^D$ | 0 |
| $f_4(x)$ | $f_4(x) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i)^2 + (x_i - 1)^2)$ | $[-10, 10]^D$ | 0 |
| $f_5(x)$ | $f_5(x) = -20e^{-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}} - e^{\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)} + 20 + e$ | $[-32, 32]^D$ | 0 |
| $f_6(x)$ | $f_6(x) = \sum_{i=1}^D (x_i + 0.5)^2$ | $[-100, 100]^D$ | 0 |
| $f_7(x)$ | $f_7(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $ | $[-10, 10]^D$ | 0 |
| $f_8(x)$ | $f_8(x) = \sum_{i=1}^D (10^6)^{i-1/D-1} x_i^2$ | $[-100, 100]^D$ | 0 |

Çizelge 7.1 ile verilen karşılaştırma fonksiyonları optimizasyon algoritmalarının performanslarının karşılaştırılmasında sıklıkla kullanılır. Bu fonksiyonların her birinin çözümü birer minimum noktası bulma problemidir ve dolayısıyla optimizasyon problemi olarak düşünülebilir. Çizelge 7.2 ile bu fonksiyonların çözümlerinden elde edilen sonuçlar özetlenmiştir. Buna göre fonksiyon çözümlerinde UPSO, PSO'ya göre daha üstün performans göstermiştir. Ayrıca Şekil 7.1-Şekil 7.8 yakınsama grafikleri incelendiğinde UPSO'nun sonuca (optimum değere) daha hızlı yakınsadığı görülmektedir. Fonksiyon ve grafikler MATLAB programı kullanılarak elde edilmiştir.

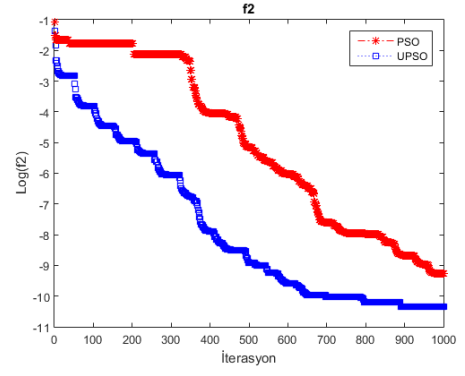
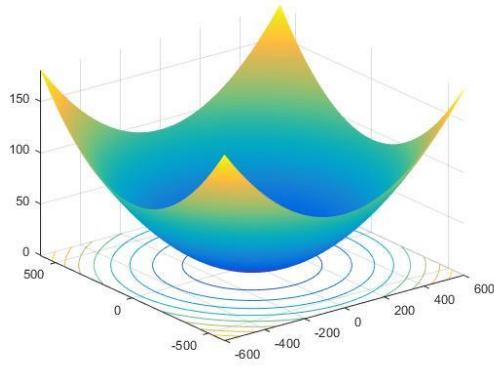
Çizelge 7.2. PSO ve UPSO performanslarının karşılaştırılması

| Fonksiyon | PSO | | UPSO | |
|-----------|------------|------------|------------|------------|
| | Ortalama | Std. Hata | Ortalama | Std. Hata |
| $f_1(x)$ | 1.7745e-04 | 1.7422e-04 | 9.0743e-05 | 6.6350e-05 |
| $f_2(x)$ | 7.9082e-05 | 7.9951e-05 | 3.6651e-05 | 2.4286e-05 |
| $f_3(x)$ | 91.3296 | 2.2842 | 16.3041 | 0.502 |
| $f_4(x)$ | 66.3325 | 2.8743 | 15.96 | 1.39 |
| $f_5(x)$ | 1.0244 | 0.0438 | 0.1148 | 0.0467 |
| $f_6(x)$ | 1.6151 | 0.1128 | 0.1128 | 1.2141e-04 |
| $f_7(x)$ | 2.8738 | 0.1125 | 0.3820 | 0.0158 |
| $f_8(x)$ | 1.4083e-14 | 4.8661e-14 | 4.5386e-16 | 1.1175e-15 |

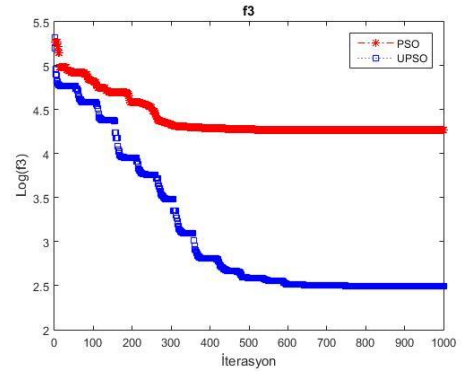
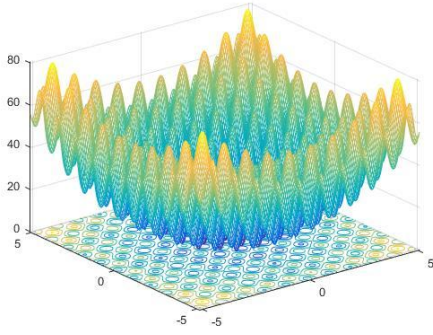
Ek olarak bu fonksiyonların grafikleri ve PSO ile UPSO kullanılarak minimum noktaya ulaşmadaki yakınsama grafikleri Şekil 7.1- Şekil 7.8 ile verilmektedir.



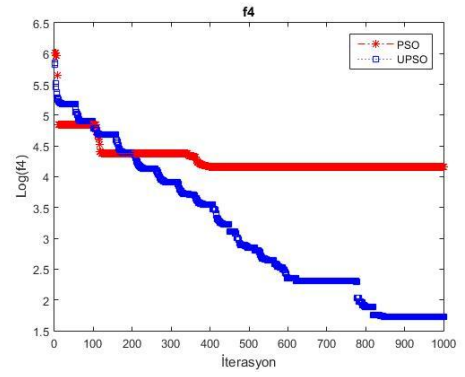
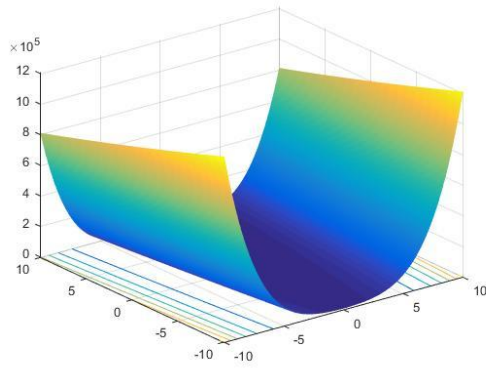
Şekil 7.1. f1 fonksiyonu ve yakınsama grafiği



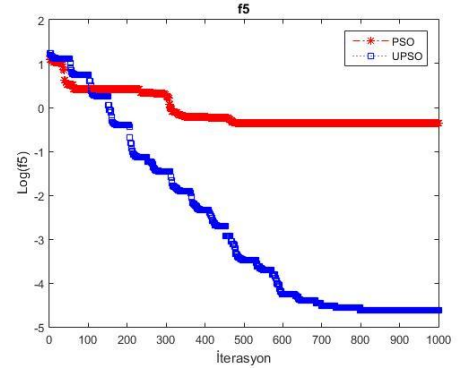
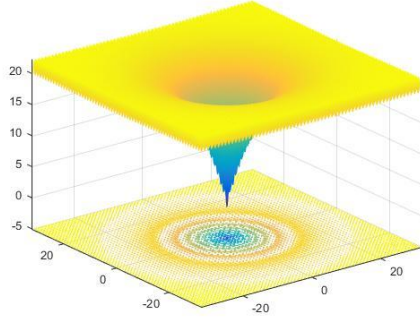
Şekil 7.2. f2 fonksiyonu ve yakınsama grafiği



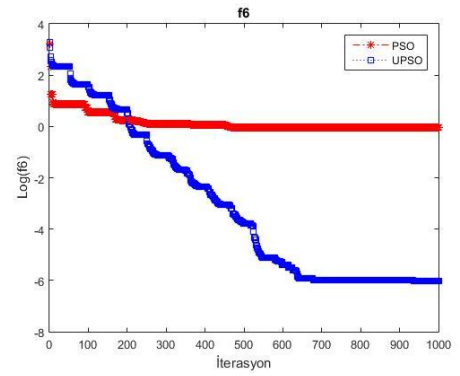
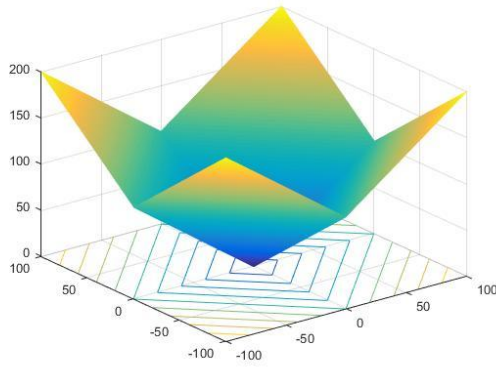
Şekil 7.3. f3 fonksiyonu ve yakınsama grafiği



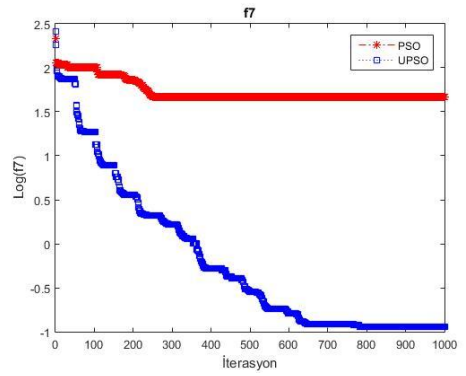
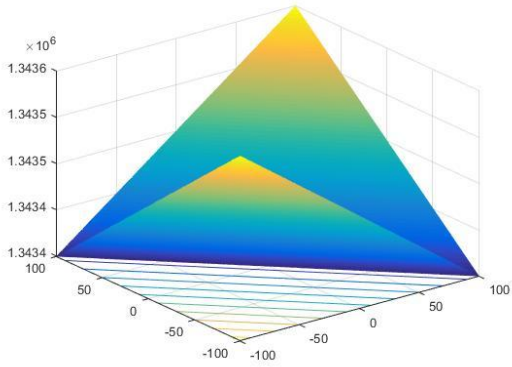
Şekil 7.4. f4 fonksiyonu ve yakınsama grafiği



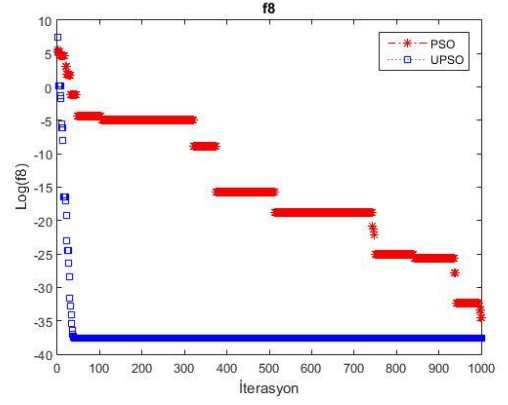
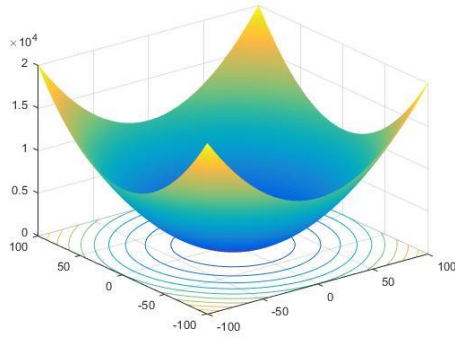
Şekil 7.5. f5 fonksiyonu ve yakınsama grafiği



Şekil 7.6. f6 fonksiyonu ve yakınsama grafiği



Şekil 7.7. f7 fonksiyonu ve yakınsama grafiği



Şekil 7.8. f8 fonksiyonu ve yakınsama grafiği

7.2. Direkt Bölme Çapraz Geçerlilik Sonuçları

Bölüm 7.1 ile UPSO'nun standart PSO'ya göre karşılaştırma fonksiyonları üzerinde genel olarak daha iyi performansa sahip olduğu gösterilmişti. Bu bölümde ise daha önce tanıtılan MLP sinir ağı, Pi-Sigma sinir ağı, Çarpımsal Nöron Model sinir ağı ve önerilen lojistik regresyon ve çarpımsal nöron modele dayalı Hibrit ağ için UPSO ile ağ eğitimleri yapılmıştır.

YSA öğrenimindeki önemli noktalardan biri de öğrenmeyi sağlayacak olan eğitim kümesinin seçilmesidir. Bu konudaki yanlış kanaatlerden birisi eğitim kümesinin ne kadar büyük seçilirse eğitmenin o kadar iyi olacağıdır. Hâlbuki eğitim kümesi en az bilgi ile en iyi öğrenmeyi sağlayacak şekilde seçilmelidir (Sönmez ve Şen, 1997). Eğitim kümesi oluşturulurken birbirine yakın bilgilerden ziyade, birbirinden farklı ve bağımsız bilgilerin seçilmesi daha verimli bir öğrenme sağlar. Aynı zamanda eğitim kümesinin daha önceki eğitilmemiş giriş değerlerine karşı uygun çıkışlar vermesi gerekir (Şen, 2004). Bu nedenle Çizelge 7.3 ile verilen sınıflandırma problemleri (Blake ve Merz, 1998) için verilerin sırasıyla %30'u, %50'si ve %70'i eğitim kümesi olarak bağımsız ve rastgele seçilmiş, kalan veriler ise geçerlilik kümesi olarak atanmıştır. Her bir sınıflandırma problemi için bağımsız şekilde 100'er deneme yapılmış ve sonuçlar her bir sınıflandırma problemi için ayrı ayrı verilmiştir. Hata fonksiyonu olarak daha iyi yakınsama sağlayan ve daha iyi sınıflandırma performansı veren çapraz entropi hata fonksiyonu (cross entropy error function, CEE) kullanılmıştır (Kline ve Berardi, 2005). Çapraz Entropi fonksiyonu

$$CEE = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) * \log(1 - \hat{y}_i)] \quad (7.1)$$

eşitliği ile verilir.

Burada y_i , i . öğrenme örneği için verilen gerçek sınıf değeri, \hat{y}_i ise modelleme sonucu elde edilen sınıf değerine ait olasılık değeridir. Literatürde farklı hata fonksiyonları da mevcuttur ancak bu tez çalışması kapsamına alınmamıştır.

Çizelge 7.3. UCI veri deposundan alınan orijinal sınıflandırma problemleri

| <i>Veri Seti</i> | <i>Bağımsız Değişken</i> | <i>Gözlem Sayısı</i> |
|--------------------------------|--------------------------|----------------------|
| <i>Avustralya Kredi Onayı</i> | 14 | 690 |
| <i>Meme Kanseri (Teşhis)</i> | 30 | 569 |
| <i>Şeker Hastalığı</i> | 8 | 768 |
| <i>Kalp Hastalığı</i> | 13 | 270 |
| <i>İyonosfer</i> | 33 | 351 |
| <i>Meme Kanseri (Orijinal)</i> | 9 | 683 |
| <i>Sağkalım Süresi</i> | 3 | 306 |

Çizelge 7.3 ile verilen sınıflandırma problemlerine ait sonuçlar değerlendirilirken dikkat edilmesi gereken önemli bir husus, verinin yüzde kaçının eğitim kümesi için seçilmesinde yeterli olup olmayacağına kanaat getirmektir. Bunun için her örnek ayrı ayrı incelenip sınıflandırma performanslarına bakarak karar vermek gerekir.

Ek olarak, bu bölümde direkt bölme çapraz geçerlilik ile önerilen yeni sınıflandırıcının performansı hakkında bilgi edinilmek amaçlanmıştır. Model seçim işlemi, direkt bölme çapraz geçerliliğe göre daha dayanıklı olan 10-kat çapraz geçerlilik yöntemi ile yapılacaktır. Hangi sınıflandırma problemi için hangi modelin seçilmesi gerektiğine Bölüm 7.3'te karar verilecektir.

- **Avustralya Kredi Onayı Veri Kümesi**

Bu veri kümesi kredi kartı uygulaması ile ilgilidir. Bu veri kümesinde 6 tanesi nümerik ve 8 tanesi kategorik olmak üzere 14 bağımsız girdi vardır.

Çizelge 7.4. Avustralya kredi onayı veri setinin eğitim performansının araştırılması

| Sınıflandırıcı | Eğitim %30 | Geçerlilik %70 | Eğitim %50 | Geçerlilik %50 | Eğitim %70 | Geçerlilik %30 |
|-----------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|
| MLP | 0,87155 (0,00198) | 0,84315 (0,00132) | 0,86525 (0,00156) | 0,84957 (0,00171) | 0,86259 (0,00111) | 0,85285 (0,00221) |
| Pi-Sigma | 0,85918 (0,00287) | 0,83108 (0,00305) | 0,85530 (0,00225) | 0,84629 (0,00252) | 0,85364 (0,00207) | 0,84686 (0,00256) |
| SMN | 0,83797 (0,00563) | 0,82575 (0,00541) | 0,81959 (0,00878) | 0,81268 (0,00899) | 0,83223 (0,00595) | 0,83045 (0,00595) |
| Hibrit (k=2) | 0,87048 (0,00212) | 0,84598 (0,00152) | 0,86475 (0,00168) | 0,85310 (0,00169) | 0,86159 (0,00104) | 0,85242 (0,00223) |
| Hibrit (k=3) | 0,86058 (0,00377) | 0,84640 (0,00329) | 0,85910 (0,00151) | 0,85270 (0,00140) | 0,85814 (0,00107) | 0,85010 (0,00219) |
| Hibrit (k=4) | 0,85995 (0,00262) | 0,84453 (0,00194) | 0,85548 (0,00171) | 0,84974 (0,00149) | 0,85614 (0,00132) | 0,85166 (0,00233) |

Tüm sınıflandırıcılar için sonuçlar incelendiğinde, eğitim kümesine verilerin %30 rastgele seçildiğinde hibrit sınıflandırıcı (k=3), verilerin %50'si eğitim kümesi için rastgele seçildiği durumda hibrit sınıflandırıcı (k=2) ve son olarak eğitim kümesi için verilerin %70'i rastgele seçildiğinde MLP sınıflandırıcısı, daha önce hiç görülmemiş verileri daha yüksek performansla sınıflandırmıştır. Bir başka deyişle, kalın punto ile gösterilen sınıflandırıcılar en iyi geçerlilik performansına sahip sınıflandırıcılardır. Bu veri kümesi için en iyi test performansı verilerin %50'si eğitim kümesi için seçildiğinde hibrit sınıflandırıcı (k=2) ile sağlanmıştır. Aynı zamanda veri kümesinde eğitim kümesinin %30 olarak seçilmesi çok fazla fark yaratmamaktadır. Bu nedenle daha az bilgi kullanılarak da bu problem için sınıflandırma yapılabileceği söylenebilir.

- **Meme Kanseri Teşhisi Veri Kümesi**

Bu veri setinde girdiler, meme kitlelerinin ince iğne aspiratlarının görüntülerinin dijitalleştirilmesinden elde edilmiştir. Kanseri teşhisinin bu görüntüleme sistemi ile ne kadar doğru tespit edileceği araştırılmıştır. Görüntüleme ile meme kitlelerinden alınan 30 özellik (girdi) ile sınıflandırma yapılmıştır.

Çizelge 7.5 incelendiğinde meme kanseri teşhisi veri kümesi için en iyi sınıflandırıcının hibrit sınıflandırıcı olduğu görülmektedir. Bu sınıflandırıcı kullanılarak meme kanseri teşhisi için yüksek doğru sınıflandırma performansı ile ağ başarılı bir şekilde eğitilebilmektedir. Veri kümesinin hiç görmediği veriler üzerinde test yapıldığında da yüksek sınıflandırma performansına sahip olduğu görülmektedir. Hibrit sınıflandırıcı eğitim kümesi yüzde kaç oranında seçilirse seçilsin diğer yöntemlerden üstün sonuçlar vermektedir. Ayrıca en yüksek geçerlilik performansı eğitim kümesinin %30 alınması ile elde edilmiştir. Bu veri seti için az bilgi kullanarak dahi başarılı sınıflandırma yapılması da hibrit metodu diğer yöntemlere göre üstün kılmaktadır.

Çizelge 7.5. Meme kanseri teşhisi veri setinin eğitim performansının araştırılması

| Sınıflandırıcı | Eğitim %30 | Geçerlilik %70 | Eğitim %50 | Geçerlilik %50 | Eğitim %70 | Geçerlilik %30 |
|-----------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|
| MLP | 0,96643 (0,00193) | 0,93219 (0,00167) | 0,95803 (0,00165) | 0,94011 (0,00203) | 0,95766 (0,00162) | 0,94339 (0,00256) |
| Pi-Sigma | 0,93649 (0,00287) | 0,91399 (0,00300) | 0,93094 (0,00272) | 0,92238 (0,00288) | 0,92240 (0,00341) | 0,92807 (0,00278) |
| SMN | 0,91138 (0,00295) | 0,88067 (0,00277) | 0,90199 (0,00286) | 0,88884 (0,00311) | 0,90039 (0,00265) | 0,89145 (0,00276) |
| Hibrit (k=2) | 0,96688 (0,00147) | 0,95550 (0,00174) | 0,97148 (0,00123) | 0,95439 (0,00148) | 0,97842 (0,00170) | 0,94902 (0,00147) |
| Hibrit (k=3) | 0,97416 (0,00167) | 0,94624 (0,00162) | 0,96486 (0,00146) | 0,95180 (0,00147) | 0,96293 (0,00122) | 0,95478 (0,00185) |
| Hibrit (k=4) | 0,98073 (0,00108) | 0,96429 (0,00108) | 0,94802 (0,00246) | 0,93883 (0,00273) | 0,94257 (0,00217) | 0,93431 (0,00244) |

- **Şeker Hastalığı Veri Kümesi**

Bu veri seti, şeker hastalığının teşhis verisidir ve veride 768 kişiye ait 8 adet özellik verilmiştir. Veri kümesi kayıp gözlem içermemektedir.

Şeker hastalığı veri setine ilişkin sınıflandırma performanslarının incelendiği Çizelge 7.6 dikkate alındığında, hibrit yöntemin (k=2) diğer yöntemlere göre daha yüksek sınıflandırma performansına sahip olduğu görülmektedir. Hibrit yöntem için k değerinin 2 alınması durumunda ağı eğitimi için ayarlanması gereken ağ parametreleri sayısı diğer yöntemlere göre daha az olacağı için (çarpımsal nöron model hariç) daha hızlı bir ağ eğitimi gerçekleştirmek mümkündür. MLP ile Hibrit yöntem karşılaştırıldığında, eğitim performansı açısından MLP'nin daha başarılı olduğu ancak veri kümesinin hiç görmediği veriler üzerinde Hibrit yöntemin daha başarılı olduğu görülmektedir.

Çizelge 7.6. Şeker hastalığı veri setinin eğitim performansının araştırılması

| Sınıflandırıcı | Eğitim %30 | Geçerlilik %70 | Eğitim %50 | Geçerlilik %50 | Eğitim %70 | Geçerlilik %30 |
|-----------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|
| MLP | 0,77439 (0,00271) | 0,73771 (0,00221) | 0,76568 (0,00197) | 0,74880 (0,00223) | 0,76493 (0,0179) | 0,74935 (0,0025) |
| Pi-Sigma | 0,76561 (0,00290) | 0,73784 (0,00270) | 0,75795 (0,00231) | 0,74318 (0,00264) | 0,76553 (0,00180) | 0,74883 (0,00271) |
| SMN | 0,70874 (0,00456) | 0,69217 (0,00439) | 0,70669 (0,00406) | 0,70190 (0,00473) | 0,71084 (0,00431) | 0,70887 (0,00509) |
| Hibrit (k=2) | 0,76581 (0,00274) | 0,74344 (0,00233) | 0,76434 (0,00217) | 0,75217 (0,00221) | 0,76140 (0,00191) | 0,75195 (0,00339) |
| Hibrit (k=3) | 0,76193 (0,00330) | 0,73960 (0,00248) | 0,75535 (0,00267) | 0,74276 (0,00225) | 0,75483 (0,00219) | 0,74909 (0,00264) |
| Hibrit (k=4) | 0,73464 (0,00331) | 0,71827 (0,00301) | 0,73615 (0,00304) | 0,72917 (0,00250) | 0,73657 (0,00236) | 0,73177 (0,00305) |

- **Kalp Hastalığı Veri Kümesi**

Bu veri kümesi kalp hastalığı ile ilişkilidir. Veri kümesinde 270 gözlem ve 13 girdi mevcuttur ve kayıp gözlem içermemektedir.

Çizelge 7.7 incelendiğinde en iyi geçerlilik performansının hibrit sınıflandırıcı ile verinin %70'inin eğitilmesi ile elde edildiği görülmektedir.

Çizelge 7.7. Kalp hastalığı veri setinin eğitim performansının araştırılması

| Sınıflandırıcı | Eğitim %30 | Geçerlilik %70 | Eğitim %50 | Geçerlilik %50 | Eğitim %70 | Geçerlilik %30 |
|-----------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|
| MLP | 0,88457 (0,00445) | 0,74989 (0,00382) | 0,83817 (0,00281) | 0,77732 (0,00334) | 0,83249 (0,00230) | 0,79827 (0,00478) |
| Pi-Sigma | 0,83539 (0,00478) | 0,72802 (0,00525) | 0,82048 (0,00357) | 0,76327 (0,00403) | 0,81832 (0,00259) | 0,78571 (0,00459) |
| SMN | 0,76708 (0,00477) | 0,71387 (0,00545) | 0,74711 (0,00507) | 0,71654 (0,00484) | 0,74635 (0,00402) | 0,72648 (0,00618) |
| Hibrit (k=2) | 0,85958 (0,00414) | 0,77046 (0,00307) | 0,84200 (0,00268) | 0,79274 (0,00301) | 0,83539 (0,00208) | 0,80412 (0,00367) |
| Hibrit (k=3) | 0,84555 (0,00473) | 0,76686 (0,00333) | 0,83976 (0,00279) | 0,78806 (0,00378) | 0,82522 (0,00206) | 0,79976 (0,00379) |
| Hibrit (k=4) | 0,81628 (0,00515) | 0,76833 (0,00360) | 0,80878 (0,00359) | 0,77449 (0,00352) | 0,80021 (0,00306) | 0,78247 (0,00415) |

- **İyonosfer Veri Kümesi**

Veri kümesi iyonosferden alınan ölçümlerle alakalı olarak gelen sinyallerin iyi ve kötü olarak sınıflandırılması ile oluşturulmuştur. Veri kümesinde 34 bağımsız girdi mevcuttur.

Çizelge 7.8 incelendiğinde MLP sınıflandırıcısının bu problem için daha iyi bir sınıflandırıcı olduğu söylenebilir.

Çizelge 7.8. İyonosfer veri setinin eğitim performansının araştırılması

| Sınıflandırıcı | Eğitim %30 | Geçerlilik %70 | Eğitim %50 | Geçerlilik %50 | Eğitim %70 | Geçerlilik %30 |
|-----------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|
| MLP | 0,92486 (0,00327) | 0,82533 (0,00406) | 0,90720 (0,00298) | 0,84159 (0,00437) | 0,89644 (0,00347) | 0,85595 (0,00437) |
| Pi-Sigma | 0,88446 (0,00376) | 0,78069 (0,00372) | 0,85585 (0,00333) | 0,79428 (0,00343) | 0,84856 (0,00283) | 0,79213 (0,00446) |
| SMN | 0,80487 (0,00708) | 0,74697 (0,00467) | 0,78455 (0,00642) | 0,75923 (0,00551) | 0,78030 (0,00531) | 0,74559 (0,00687) |
| Hibrit (k=2) | 0,86366 (0,00575) | 0,78902 (0,00353) | 0,86472 (0,00458) | 0,80354 (0,00372) | 0,85286 (0,00348) | 0,81273 (0,00410) |
| Hibrit (k=3) | 0,85772 (0,00431) | 0,78690 (0,00414) | 0,85623 (0,00368) | 0,79997 (0,00440) | 0,83180 (0,00392) | 0,81236 (0,00503) |
| Hibrit (k=4) | 0,83830 (0,00494) | 0,78096 (0,00360) | 0,81397 (0,00451) | 0,78769 (0,00414) | 0,79268 (0,00536) | 0,78223 (0,00545) |

- **Meme Kanseri (Orijinal) Veri Kümesi**

Bu veri seti, toplam 9 girdi ve 699 gözlemden oluşmaktadır. Veri kümesi kayıp gözlem içermektedir. Bu gözlemler göz ardı edilerek veri kümesinden çıkarılmış ve 683 gözlem üzerinden sınıflandırma performansı araştırılmıştır.

Çizelge 7.9 incelendiğinde hibrit yöntemin daha iyi kestirim gücüne sahip olduğu söylenebilir.

Çizelge 7.9. Meme kanseri veri setinin eğitim performansının araştırılması

| Sınıflandırıcı | Eğitim %30 | Geçerlilik %70 | Eğitim %50 | Geçerlilik %50 | Eğitim %70 | Geçerlilik %30 |
|-----------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|
| MLP | 0,98502 (0,00120) | 0,95862 (0,00114) | 0,97766 (0,00095) | 0,96416 (0,00107) | 0,97495 (0,00069) | 0,96568 (0,00148) |
| Pi-Sigma | 0,96137 (0,00159) | 0,95228 (0,00144) | 0,96525 (0,00120) | 0,95664 (0,00133) | 0,96186 (0,00137) | 0,95580 (0,00182) |
| SMN | 0,90732 (0,00337) | 0,90035 (0,00339) | 0,89592 (0,00640) | 0,89128 (0,00654) | 0,89401 (0,00451) | 0,89034 (0,00568) |
| Hibrit (k=2) | 0,98141 (0,00112) | 0,96414 (0,00081) | 0,97724 (0,00078) | 0,96599 (0,00073) | 0,97571 (0,00065) | 0,96805 (0,00118) |
| Hibrit (k=3) | 0,97907 (0,00105) | 0,96521 (0,00085) | 0,97654 (0,00073) | 0,96690 (0,00080) | 0,9724 (0,00070) | 0,96702 (0,00117) |
| Hibrit (k=4) | 0,97420 (0,00130) | 0,96381 (0,00083) | 0,97238 (0,00088) | 0,96599 (0,00089) | 0,97054 (0,00072) | 0,96444 (0,00122) |

- **Sağkalım Süresi Veri Kümesi**

Bu veri seti meme kanseri ameliyatı olan hastaların hayatta kalma sürelerine ilişkindir. Hastaların hayatta kalma süreleri, 5 yıldan içinde ölmesi ya da 5 yıldan daha fazla yaşama durumlarına göre sınıflandırılmıştır. Veri setinde 3 bağımsız girdi ve 306 gözlem mevcuttur.

Çizelge 7.10 incelendiğinde SMN ve Hibrit modelin performanslarının birbirlerine çok yakın olduğu ve veri kümesinin %50'sinin geçerlilik kümesi olarak alındığında en iyi kestirim performansına ulaşıldığı söylenebilir.

Çizelge 7.10. Sağkalım süresi veri setinin eğitim performansının araştırılması

| Sınıflandırıcı | Eğitim %30 | Test %70 | Eğitim %50 | Test %50 | Eğitim %70 | Test %30 |
|-----------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|
| MLP | 0,77241 (0,00347) | 0,73634 (0,00246) | 0,76863 (0,00261) | 0,74105 (0,00246) | 0,76210 (0,00207) | 0,73957 (0,00399) |
| Pi-Sigma | 0,77283 (0,00437) | 0,73491 (0,00221) | 0,75353 (0,00287) | 0,73536 (0,00266) | 0,75827 (0,00206) | 0,74033 (0,00469) |
| SMN | 0,76435 (0,00357) | 0,73350 (0,00210) | 0,74725 (0,00251) | 0,74346 (0,00248) | 0,74738 (0,00197) | 0,73848 (0,00433) |
| Hibrit (k=2) | 0,75705 (0,00389) | 0,73782 (0,00203) | 0,75595 (0,00267) | 0,74281 (0,00271) | 0,75481 (0,00156) | 0,74185 (0,00370) |
| Hibrit (k=3) | 0,76467 (0,00401) | 0,73939 (0,00207) | 0,76046 (0,00269) | 0,73915 (0,00259) | 0,75425 (0,00180) | 0,73978 (0,00387) |
| Hibrit (k=4) | 0,75620 (0,00430) | 0,73916 (0,00204) | 0,75353 (0,00287) | 0,73536 (0,00266) | 0,74963 (0,00201) | 0,73337 (0,00398) |

7.3. 10-kat Çapraz Geçerlilik Sonuçları

Bu bölümde 10-kat çapraz geçerlilik yöntemi sınıflandırma problemlerine uygulanarak hangi problem için hangi modelin seçilmesi gerektiğine karar verilmiştir.

Çizelge 7.11 incelendiğinde hibrit sınıflandırıcının (k=2) diğer sınıflandırıcılara göre daha iyi kestirim performansı gösterdiği görülmektedir. Bu

nedenle bu sınıflandırma Avustralya kredi onayı sınıflandırma problemi için hibrit sınıflandırıcı diğer sınıflandırıcılara tercih edilir.

Çizelge 7.11. Avustralya kredi onayı problemi için 10-kat çapraz geçerlilik

| 10 kat Çapraz Geçerlilik | |
|--------------------------|-----------------|
| Sınıflandırıcı | Hata Ortalaması |
| MLP | 0,1404 |
| Pi-Sigma | 0,1595 |
| SMN | 0,1608 |
| Hibrit (k=2) | 0,1318 |
| Hibrit (k=3) | 0,1436 |
| Hibrit (k=4) | 0,1421 |

Meme kanseri teşhisi sınıflandırma problemi için 10-kat çapraz geçerlilik sonuçları Çizelge 7.12’de verilmiştir. Çizelge 7.12 incelendiğinde meme kanseri teşhisi sınıflandırma problemi için en küçük kestirim hatası ortalaması yine hibrit sınıflandırıcı ile elde edilmiştir. Bu nedenle bu sınıflandırma problemi için hibrit yöntemin daha iyi bir sınıflandırıcı olduğu söylenebilir. Aynı zamanda MLP yöntemi kullanılarak elde edilen performans, Pi-Sigma ve SMN ağlarına göre daha iyi performans göstermektedir.

Çizelge 7.12. Meme kanseri teşhisi problemi için 10-kat çapraz geçerlilik

| 10 kat Çapraz Geçerlilik | |
|--------------------------|-----------------|
| Sınıflandırıcı | Hata Ortalaması |
| MLP | 0,05230 |
| Pi-Sigma | 0,1057 |
| SMN | 0,0963 |
| Hibrit (k=2) | 0,03684 |
| Hibrit (k=3) | 0,0456 |
| Hibrit (k=4) | 0,05443 |

Çizelge 7.13 incelendiğinde şeker hastalığı sınıflandırma problemi için en iyi sınıflandırıcının hibrit sınıflandırıcı olduğu görülmektedir. Hibrit sınıflandırıcı dışında bu sınıflandırma problemi için Pi-Sigma sinir ağı sınıflandırıcı ile diğer sınıflandırıcılara göre daha düşük hatalı sınıflandırma performansı elde edilmiştir.

Çizelge 7.13. Şeker hastalığı problemi için 10-kat çapraz geçerlilik

| 10 kat Çapraz Geçerlilik | |
|--------------------------|-----------------|
| Sınıflandırıcı | Hata Ortalaması |
| MLP | 0,2655 |
| Pi-Sigma | 0,2397 |
| SMN | 0,2787 |
| Hibrit (k=2) | 0,2265 |
| Hibrit (k=3) | 0,2513 |
| Hibrit (k=4) | 0,2422 |

Çizelge 7.14 incelendiğinde yine hibrit sınıflandırıcı diğer yöntemlere göre daha iyi kestirim performansı göstermektedir. Hibrit sınıflandırıcı dışındaki diğer üç yöntem arasındaki sıralama ise Pi-Sigma, MLP ve SMN sinir ağı şeklindedir.

Çizelge 7.14. Kalp hastalığı problemi için 10-kat çapraz geçerlilik

| 10 kat Çapraz Geçerlilik | |
|--------------------------|-----------------|
| Sınıflandırıcı | Hata Ortalaması |
| MLP | 0,2185 |
| Pi-Sigma | 0,2000 |
| SMN | 0,2259 |
| Hibrit (k=2) | 0,1815 |
| Hibrit (k=3) | 0,1926 |
| Hibrit (k=4) | 0,2037 |

Çizelge 7.15 incelendiğinde meme kanseri (orijinal) sınıflandırma problemi için hibrit model daha iyi bir sınıflandırıcıdır. Ek olarak hibrit modelde farklı k değerleri ile elde edilen sonuçların hepsi diğer yöntemlere üstünlük sağlamıştır.

Çizelge 7.15. Meme kanseri (orijinal) problemi için 10-kat çapraz geçerlilik

| 10 kat Çapraz Geçerlilik | |
|--------------------------|-----------------|
| Sınıflandırıcı | Hata Ortalaması |
| MLP | 0,03894 |
| Pi-Sigma | 0,04093 |
| SMN | 0,0937 |
| Hibrit (k=2) | 0,03220 |
| Hibrit (k=3) | 0,02495 |
| Hibrit (k=4) | 0,03652 |

Çizelge 7.16 incelendiğinde iyonosfer iyi-kötü sinyal sınıflandırma problemi için MLP sinir ağı sınıflandırıcısının diğer yöntemlerden daha üstün olduğu görülmektedir.

Çizelge 7.16. İyonosfer problemi için 10-kat çapraz geçerlilik

| Sınıflandırıcı | 10 kat Çapraz Geçerlilik Hata Ortalaması |
|----------------|---|
| MLP | 0,1400 |
| Pi-Sigma | 0,1880 |
| SMN | 0,3048 |
| Hibrit (k=2) | 0,1855 |
| Hibrit (k=3) | 0,2222 |
| Hibrit (k=4) | 0,1989 |

Sağkalım süresi sınıflandırma problemine ait 10-kat çapraz geçerlilik sonuçları Çizelge 7.17’de gösterilmiştir. Çizelge 7.17’ye göre bu sınıflandırma problemi için en iyi sınıflandırıcının hibrit sınıflandırıcı olduğu sonucuna varılmıştır.

Çizelge 7.17. Sağkalım süresi problemi için 10-kat çapraz geçerlilik

| Sınıflandırıcı | 10 kat Çapraz Geçerlilik Hata Ortalaması |
|----------------|---|
| MLP | 0,2647 |
| Pi-Sigma | 0,2547 |
| SMN | 0,2578 |
| Hibrit (k=2) | 0,2582 |
| Hibrit (k=3) | 0,2549 |
| Hibrit (k=4) | 0,2486 |

8. SONUÇ VE ÖNERİLER

İstatistiksel sınıflandırma problemleri, bilimin her alanında karşılaşılabilecek bir problem türüdür. Bu tip problemlerin çözümü için mevcut yöntemler üzerinde iyileştirilme çalışmalarına ve yeni yöntem arayışları devam etmektedir. Son yıllarda sınıflandırma problemlerinin çözümünde YSA'lar sıklıkla kullanılmaktadır. YSA'ların sınıflandırma becerilerinin diğer yöntemlerden daha üstün olması, yöntemin kullanımında herhangi bir varsayım olmaması, lineer olmayan sistemlerin çözümünde de etkin olması ve yeni gözlemleri yüksek performansla sınıflandırabilmesi gibi özellikleri YSA'yı tercih edilebilir kılmaktadır. Literatürde en bilindik YSA çeşidi MLP'dir. MLP'de katman sayısına ve katmanlardaki hücre sayısına karar verme gibi zorlukların olması ve yakınsama süresinin fazla olması, araştırmacıları yeni sinir ağı arayışlarına itmiştir. Pi-Sigma ve SMN sinir ağıları, MLP'ye alternatif olarak önerilmiştir.

YSA'ların başarılı sonuçlar vermesi ancak başarılı bir ağ eğitimi ile mümkün olabilmektedir. Bu nedenle YSA'ların eğitimi için literatürde onlarca eğitim algoritması kullanılmıştır. Bu algoritmalarından birisi de evrimsel algoritmalarından bir tanesi olan PSO'dur. Aynı zamanda PSO'nun yakınsama hızının artırılması ve optimum noktaya daha az hata ile yaklaşabilmesi için PSO üzerinde çalışmalar yapılmıştır.

Bu tez çalışmasında literatürde kullanılan MLP, Pi-Sigma ve SMN ağılarına alternatif olarak lojistik regresyon ve çarpımsal nöron modele dayalı yeni bir YSA sınıflandırıcısı önerilmiştir. Önerilen bu sınıflandırıcı ağının eğitimi için UPSO önerilmiştir. UPSO'nun performansının PSO'dan daha iyi olduğunu göstermek için matematikte sıklıkla kullanılan karşılaştırma fonksiyonları ile karşılaştırma yapılmıştır ve UPSO'nun genel olarak daha iyi performans gösterdiğine karar verilmiştir.

Önerilen yeni yöntemin sınıflandırma performansının araştırılması için literatürde sıklıkla kullanılan gerçek hayat problemleri "UCI Irvine Machine

Learning Repository” (Blake ve Merz, 1998) veri deposundan alınarak incelenmiştir. Tez çalışmasında adı geçen sinir ağlarının eğitim ve geçerlilik performanslarının araştırılması için değişik oranlarda eğitim ve geçerlilik kümeleri seçilerek sınıflandırıcıların performansları incelenmiş ve sonuçlar Bölüm 7.2’de özetlenmiştir.

Önerilen yöntem ve diğer mevcut yöntemler arasında hangi yöntemin tercih edilmesi gerektiğine ise Bölüm 6’da verilen 10-kat çapraz geçerlilik yöntemi ile karar verilmiştir. Çapraz Geçerlilik ile hangi sınıflandırma problemleri için hangi sınıflandırıcının daha yüksek geçerlilik performansına sahip olduğu incelenmiştir.

Elde edilen sonuçlara göre iyonosfer sınıflandırma problemi dışında tüm sınıflandırma problemleri için hibrit sınıflandırıcının diğer yöntemlere göre daha iyi sınıflandırma performansına sahip olduğuna karar verilmiştir.

Hibrit yöntemde k sayısına karar verilmesi yöntemin bir handikapı olarak düşünülebilir. Hibrit yöntem için k sayısının artması, ağdaki parametre sayısının artmasına ve matematiksel modelin daha karmaşık bir hal almasına neden olabilmektedir. Parametre sayısındaki artış, sınıflandırıcının geç yakınsamasına ve eğitim süresini uzamasına neden olabilir. Bu nedenle k değeri mümkün olduğunca küçük seçilmeye çalışılmalıdır.

KAYNAKLAR

- Atakan, C. (1997). Diskriminasyon ve hata oranı tahminleri. Doktora Tezi (basılmamış), Ankara Üniversitesi Fen Bilimleri Enstitüsü İstatistik Anabilim Dalı, 123, Ankara.
- Bansal, J. C., Singh, P. K., Saraswat, M., Verma, A., Jadon, S. S. and Abraham, A. 2011. Inertia weight strategies in particle swarm optimization. 2011 Third World Congress on Nature and Biologically Inspired Computing (NaBIC), 640-647. DOI: 10.1109/NaBIC.2011.6089659
- Baş, E., Uslu, V. R. and Eğrioğlu, E. 2016. Robust learning algorithm for multiplicative neuron model artificial neural networks. Expert Systems with Applications, 56, 80-88. DOI: 10.1016/j.eswa.2016.02.051
- Başarrır, G. (1990). Çok değişkenli verilerde ayırimsama sorunu ve lojistik regresyon analizi. Doktora Tezi (basılmamış), Hacettepe Üniversitesi Fen Bilimleri Enstitüsü İstatistik Anabilim Dalı, 145, Ankara.
- Becker, S. and Cun, Y. L. 1988. Improving the convergence of back-propagation learning with second order methods, Tech. Rep. CRG-TR-88-5, Connectionist Research Group, University of Toronto, Canada.
- Bergh, F. van den and Engelbrecht, A.P. 2001. Effect of swarm size on Cooperative Particle Swarm Optimisers. Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation. 892-899, San Francisco, California.
- Bishop, C. M. 2006. Pattern recognition and machine learning. Springer, 738, USA.
- Blake, C. L. and Merz, C. J. 1998. UCI Repository of machine learning databases, Univ. Calif. (1998) <http://archive.ics.uci.edu/ml/>.
- Brent, R. P. 1991. Fast training algorithms for multi-layer neural nets. IEEE Transactions on Neural Networks, 2, 346-354. DOI: 10.1109/72.97911
- Browne, M. W. 2000. Cross-Validation Methods. Journal of Mathematical Psychology, 44, 108-132. DOI:10.1006/jmps.1999.1279
- Burse, K., Manoria, M. and Kirar, V. P. S. 2011. Improved back propagation algorithm to avoid local minima in multiplicative neuron model. Information Technology and Mobile Communication, 67-73. DOI: 10.1007/978-3-642-20573-6_11
- Castillo, P. A., Merelo, J. J., Prieto, A., Rivas, R. and Romero, G. 2000. G-Prop: Global optimization of multilayer perceptrons using GAs. Neurocomputing, 35, 149-163. DOI: 10.1016/S0925-2312(00)00302-7

- Chan, C. L. and Chen, C. L. 2015. A cautious PSO with conditional random. *Expert Systems with Applications*, 42, 4120-4125. DOI: 10.1016/j.eswa.2014.12.046
- Cheng, S. and Pecht, M. 2012. Using cross-validation for model parameter selection of sequential probability ratio test. *Expert System with Applications*, 39,8467-8473. DOI: 10.1016/j.eswa.2012.01.172
- Clerc, M. 1999. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. *Proceedings of International Conference on Electronic Commerce, Washington, 1951-1957.* DOI: 10.1109/CEC.1999.785513
- Cui, H., Feng, J., Guo, J. and Wang, T. 2015. A novel single multiplicative neuron model trained by an improved glowworm swarm optimization algorithm for time series prediction. *Knowledge-Based Systems*, 88, 195-209. DOI:10.1016/j.knosys.2015.07.032
- Czepiel, S. A. 2002. Maximum likelihood estimation of logistic regression models: theory and implementation. <http://czep.net/stat/mlelr.pdf> (Erişim tarihi: 20.05.2016)
- Da, Y. and Xiurun, G. 2005. An improved PSO-based ANN with simulated annealing technique. *Neurocomputing*, 63, 527-533. DOI: 10.1016/j.neucom.2004.07.002
- Eberhart, R. C. and Kennedy, J. 1995. A new optimizer using particle swarm theory. *Proc. sixth international symposium on micro machine and human science (Nagoya, Japan), IEEE Services Center, Piscataway, NJ*, 39-43. DOI: 10.1109/MHS.1995.494215
- Eberhart, R. C. and Shi, Y. 2000. Comparing inertia weights and constriction factors in particle swarm optimization. *Proceedings of the 2000 Congress on Evolutionary Computaitaion*, 84-88. DOI: 10.1109/CEC.2000.870279
- Efron, B. 1983. *The Jackknife, the Bootstrap and other resampling plans (Second edition)*. SIAM, 103, Philadelphia.
- Elmas, Ç. 2011. *Yapay zeka uygulamaları (İkinci Baskı)*, Seçkin Yayıncılık, 424, Ankara.
- Feng, Y., Teng, G. F., Wang, A. X. and Yao, Y. M. 2007. Chaotic Inertia Weight in Particle Swarm Optimization. In *Proceedings of the IEEE international conference on innovative computing, information and control*,475-479, Washington DC: IEEE Computer Society. DOI: 10.1109/ICICIC.2007.209
- Fisher, R. A. 1936. The use of multiple measurement in Taxonomic problems. *Annals of Eugenics*, 7, 179-188. DOI: 10.1111/j.1469-1809.1936.tb02137.x
- Geng, H., Huang, Y., Gao, J. and Zhu, H. 2013. A self-guided Particle Swarm Optimization with Independent Dynamic Inertia Weights Setting on Each Particle. *Applied Mathematics ve Information Sciences*, 7(2), 545-552. DOI: 10.12785/amis/070217

- Ghazali, R. and al-Jumeily. 2009. Application of Pi-Sigma Neural Networks and Ridge Polynomial Neural Networks to Financial Time Series Prediction. Artificial Higher Order Neural Networks for Economics and Business Hershey, New York: IGI Global. 271-293.
- Gündođdu, Ö., Eđriođlu, E., Aladađ, Ç. H. and Yolcu, U. 2015. Multiplicative neuron model artificial neural network based on Gaussian activation function. Neural Computing and Applications, 927-935. DOI: 10.1007/s00521-015-1908-x
- Hagan, M. T. and Menhaj, M. B. 1994. Training feedforward networks with the Marquardt algorithm. IEEE Transactions on Neural Networks, 5, 989-993. DOI: 10.1109/72.329697
- Hagan, M. T., Demuth, H. B. and Beale, M. 1996. Neural Network Design. PWS Publishing Company, 734, United State of America.
- Hebb, D. O. 1949. The organization of behavior: a neuropsychological theory. Wiley, 335, Newyork.
- Hosmer, D. W. and Lemeshow, S. 2000. Applied logistic regression (Second Edition). Wiley Series in Probability and Statistics, 396, United States of America.
- Huang, T. and Mohan, A. S. 2006. Micro-particle swarm optimizer for solving high dimensional optimization problems. Applied Mathematics and Computation, 181, 1148-1154. DOI:10.1016/j.amc.2006.01.088
- Hussain, A. J. and Liatsis, P. 2002. Recurrent pi-sigma networks for DPCM image coding. Neurocomputing, 55, 363-382. DOI: 10.1016/S0925-2312(02)00629-X
- Ilonen, J., Kamarainen, J. K. and Lampinen, J. 2003. Differential evolution training algorithm for feed-forward neural networks. Neural Processing Letters, 17, 93-105. DOI: 10.1023/A:1022995128597
- Izenmann, A. J. 2008. Modern multivariate statistical techniques. Springer, 237-243, USA.
- Jacobs, R. A. 1988. Increased rates of convergence through learning rate adaptation. Neural Networks, 295-307.
- Jain, A. K., Mao, J. and Mohiuddin, K. M. 1996. Artificial neural networks: a tutorial. IEEE Computer, 31-44. DOI: 10.1109/2.485891
- Johansson, E. M., Dowla, F. U. and Goodman, D. M. 1992. Backpropagation learning for multilayer feedforward neural Networks using conjugate gradient method. International Journal of Neural Systems, 2(4), 291-301. DOI: 10.1142/S0129065791000261
- Juang, C. F. 2004. A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. IEEE Transactions on Systems, 34(2), 997-1006. DOI: 10.1109/TSMCB.2003.818557

- Karaboğa, D. and Basturk, B. 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39, 459-471. DOI: 10.1007/s10898-007-9149-x
- Kline, M. D. and Berardi, V. L. 2005. Revisiting squared-error and cross-entropy functions for training neural network classifiers. *Neural Computing ve Applications*, 14, 310-318. DOI: 10.1007/s00521-005-0467-y
- Kohavi, R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1137-1143, San Francisco, CA: Morgan Kaufmann.
- Kolay, E., Tunc, T. and Eğrioğlu, E. 2016. Classification with Some Artificial Neural Network Classifiers Trained a Modified Particle Swarm Optimization, *American Journal of Intelligent Systems*, 6(3), 59-65 DOI:10.5923/j.ajis.20160603.01
- Knoke, J. D. 1982. Discriminant analysis with discrete and continuous variables. *Biometrics*, 38, 191-200. DOI: 10.2307/2530302
- Kubat, C. 2012. *Matlab yapay zeka ve mühendislik uygulamaları*. Beşiz Yayınları, 694. Sakarya.
- Li, C. K. 2003. A Sigma-Pi-Sigma Neural Network (SPSNN). *Neural Processing Letters*. 17,1-19. DOI: 10.1023/A:1022967523886
- Lin, S. W. and Chen, S. C. 2009. PSOLDA: A particle swarm optimization approach for enhancing classification accuracy rate of linear discriminant analysis. *Applied Soft Computing journal*, 9, 1008-1015. DOI: 10.1016/j.asoc.2009.01.001
- Liu, Y., Wang, G., Chen, H., Dong, H., Zhu, X. and Wang, S. 2011. An improved particle swarm optimization for feature selection. *Journal of Bionic Engineering*, 8,191-200. DOI: 10.1016/S1672-6529(11)60020-6
- Liang, J. J., Qin, A. K. and Baskar, P. N. 2006. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, 10(3), 281-295. DOI: 10.1109/TEVC.2005.857610
- Liang, X., Li, W., Zhang, Y. and Zhou, M. 2015. An adaptive particle swarm optimization method based on clustering. *Soft Computing*, 19, 431-448. DOI 10.1007/s00500-014-1262-4
- Ludermir, T. B., Yamazaki, A. and Zanchettin, C. 2006. An optimization methodology for neural networks weights and architectures. *IEEE Transactions on Neural Networks*, 17(6), 1452-1459. DOI: 10.1109/TNN.2006.881047
- Malik, R. F., Rahman, T. A., Hashim, S. Z. M. and Ngah, R. 2007. New particle swarm optimizer with Sigmoid increasing inertia weight. *International Journal of Computer Science and Security (IJCSS)*, 1(2), 35-44.

- McCulloch, W. S. and Pitts, W. 1943. A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics* 5, 115-133. DOI:10.1007/BF02478259
- Mendes, R., Cortez, P., Rocha, M. and Neves, J. 2002. Particle swarms for feedforward neural network training. *Proceedings of the International Joint Conference on Neural Networks*, 1895-1899. DOI: 10.1109/IJCNN.2002.1007808
- Mendes, R., Kennedy, J. and Neves, J. 2004. The fully informed particle swarm: Simpler, maybe beter. *IEEE Transactions on Evolutionary Computation*, 8(3),204-210. DOI: 10.1109/TEVC.2004.826074
- Montana, D. J. and Davis, I. 1990. Training feedforward networks using genetic algorithms. *Machine Learning*, 762-767.
- Nayak, J., Naik, B. and Behera, H. S. 2014. A hybrid PSO-GA based Pi sigma neural network (PSNN) with standard back propagation gradient descent learning for classification. 2014 *International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, IEEE. DOI: 10.1109/ICCICCT.2014.6993082
- Örkcü, H. H. (2009). Ayırma analizine matematiksel programlama ve yapay sinir ağı yaklaşımları. Doktora Tezi (basılmamış), Gazi Üniversitesi Fen Bilimleri Enstitüsü İstatistik Anabilim Dalı, 171, Ankara.
- Öztemel, E. 2012. Yapay sinir ağları (3. Baskı), Papatya Yayıncılık, 232, İstanbul.
- Öztürk, C. 2011. Yapay sinir ağlarının yapay arı kolonisi algoritması ile eğitilmesi. Doktora Tezi (basılmamış), Erciyes Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı, 205, Kayseri.
- Panigrahi, S., Bhoi, A. K. and Karali, Y. 2013. A Modified Differential Evolution Algorithm trained Pi-Sigma Neural Network for Pattern Classification. *International Journal of Soft Computing and Engineering (IJSCE)*, 3(5), 133-136. DOI: 10.4018/978-1-59904-897-0.ch012
- Parlos, A. G., Fernandez, B., Atiya, F. B., Muthusami, J. and Tsai, J. K. 1994. An accelerated learning algorithm for multilayer perceptron networks. *IEEE Transactions on Neural Networks*, 5, 493-497. DOI: 10.1109/72.286921
- Rencher, A. C. 2002. *Methods of multivariate analysis (Second Edition)*. Wiley Series in Probability and Statistics, 300-302, United States of America.
- Rosenblatt, F. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65, 386-408.
- Rumelhart, D. E., Hinton, G. H. and Williams, R. J. 1986. Learning representation by back propagation error, in *Nature*, 32, 533-536.

- Salerno, J. 1997. Using the particle swarm optimization technique to train a recurrent neural model. IEEE International Conference on Tools with Artificial Intelligence, 45-49. DOI: 10.1109/TAI.1997.632235
- Shi, Y. and Eberhart, R. C. 1998. A Modified Particle Swarm Optimizer. IEEE International Conference on Evolutionary Computation, Alaska. DOI: 10.1109/ICEC.1998.699146
- Shi, Y. and Eberhart, R. C. 2001. Fuzzy Adaptive Particle Swarm Optimization. Proceedings of the 2001 Congress on Evolutionary Computation, 101-106. DOI: 10.1109/CEC.2001.934377
- Shin, Y. and Ghosh, J. 1991. The Pi-sigma Network : An efficient higher-order neural network for pattern classification and function approximation. Proceedings of International Joint Conference on Neural Networks, 1,13-18. DOI: 10.1109/IJCNN.1991.155142
- Suratgar, A. A., Tavakoli, M. B. and Hoseinabadi, A. 2007. Modified Levenberg–Marquardt method for neural networks training. World Academy of Science, Engineering and Technology, 6, 46-48.
- Sönmez, İ. ve Şen, Z. 1997. Yapay sinir ağları yardımıyla debi öngörüsü. Meteorolojik Karakterli Doğal Afetler Sempozyumu, 445-457, Ankara.
- Şen, Z. 2004. Yapay sinir ağları ilkeleri. Su Vakfı Yayınları, 183, İstanbul.
- Tatlıldil, H. 1996. Uygulamalı çok değişkenli istatistiksel analiz. Cem Web Ofset Ltd. Şti. (Akademi Matbaası), 424, Ankara
- Tunc, T. 2012. A New Hybrid Method Logistic Regression and Feedforward Neural Network for Lung Cancer Data. Mathematical Problems in Engineering. DOI:10.1155/2012/241690
- Van den Bergh, F. and Engelbrecht, A. P. 2004. A cooperative approach to particle swarm optimization. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, 8(3), 225-239. DOI: 10.1109/TEVC.2004.826069
- Widrow, B., Winter, R. G., and Baxter, R. A. 1988. Layered neural nets for pattern recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing, 36, 1109-1118. DOI: 10.1109/29.1638
- Wolpert, D. H. 1996. The Lack of A Priori Distinctions Between Learning Algorithms. Neural Computation, 8(7), 1341-1390. Massachusetts Institute of Technology. DOI: 10.1162/neco.1996.8.7.1341
- Wu, X., Zhu, Z., Su, X., Fan, S., Du, Z., Chang, Y. and Zeng, Q. (2015). A study of single multiplicative neuron model with nonlinear filters for hourly wind speed prediction. Energy, 88, 194-201. DOI: 10.1016/j.energy.2015.04.075

- Xin, J., Chen, G. and Hai, Y. 2009. A particle swarm optimizer with multi-stage linearly-decreasing inertia weight. 2009 International Joint Conference on Computational Sciences and Optimization, 505-508. DOI: 10.1109/CSO.2009.420
- Xu, X., Tang, Y., Li, J., Hua, C. and Guan, X. 2015. Dynamic multi-swarm particle swarm optimizer with cooperative learning strategy. Applied Soft Computing, 29, 169-183. DOI: 10.1016/j.asoc.2014.12.026
- Yadav, R. N., Kalra, P. K. and John, J. 2007. Time series prediction with single multiplicative neuron model, Applied Soft Computing, 7, 1157–1163. DOI: 10.1016/j.asoc.2006.01.003
- Yang, Y. 2007. Consistency of Cross Validation for Comparing Regression Procedures. The Annals of Statistics, 35(6), 2450-2473. DOI: 10.1214/009053607000000514
- Yolcu, U., Eğrioglu, E. and Aladağ, Ç. H. 2013. A new linear ve nonlinear artificial neural network model for time series forecasting. Decision Support Systems, 54, 1340-1347. DOI: 10.1016/j.dss.2012.12.006
- Zanchettin, C., Ludermir, T. B. and Almeida, L. M. 2011. Hybrid training method for MLP: optimization of architecture and training. IEEE Transactions on Systems, 41(4),1097-1109. DOI: 10.1109/TSMCB.2011.2107035
- Zhang, G. 2000. Neural network for classification: a survey. IEEE Transactions on Systems, 30(4), 451-462. DOI: 10.1109/5326.897072
- Zhang, J. R., Zhang, J., Lok, T. M. and Lyu, M. R. 2007. A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. Applied Mathematics and Computation, 185, 1026-1037. DOI: 10.1016/j.amc.2006.07.025
- Zhao, L. and Yang, Y. 2009. PSO-based single multiplicative neuron model for time series prediction. Expert Systems with Applications, 36(2), 2805-2812. DOI: 10.1016/j.eswa.2008.01.061

ÖZGEÇMİŞ

Adı ve Soyadı : Erdinç Kolay
Doğum Yeri : İstanbul
Doğum Tarihi : 26/12/1986
Yabancı Dil : İngilizce

Eğitim Durumu

Lise : Fenerbahçe Lisesi (2004)
Lisans : Ondokuz Mayıs Üniversitesi Sinop Fen Edebiyat Fakültesi
İstatistik Bölümü (2008)
Yüksek Lisans : Sinop Üniversitesi Fen Bilimleri Enstitüsü
İstatistik Anabilim Dalı (09.2008-08.2011)

Çalıştığı Kurum/Kurumlar

Sinop Üniversitesi İstatistik Bölümü, Araştırma Görevlisi

Yayınlar

E. Kolay, T. Tunç, E. Eğrioğlu, Classification with Some Artificial Neural Network Classifiers Trained a Modified Particle Swarm Optimization, American Journal of Intelligent Systems, 6 (2016) 59–65.
doi:10.5923/j.ajis.20160603.01.